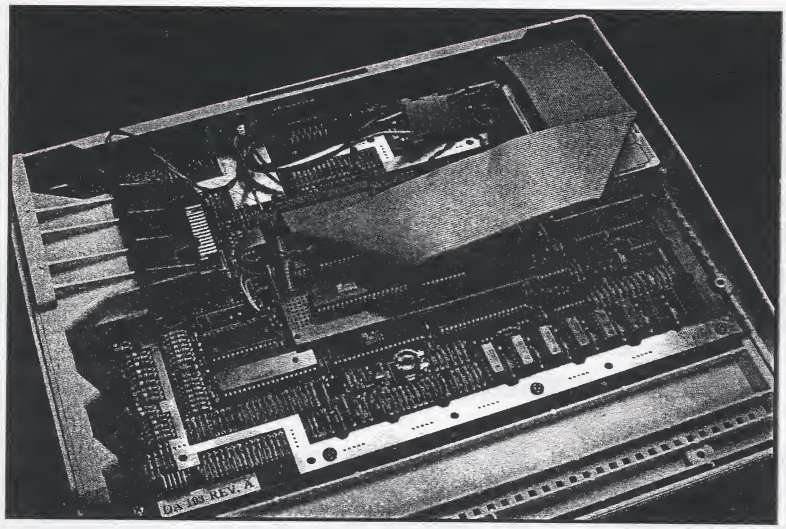
# Atari Classics

Fall 1996 $4.95

Volume 5, Number 2

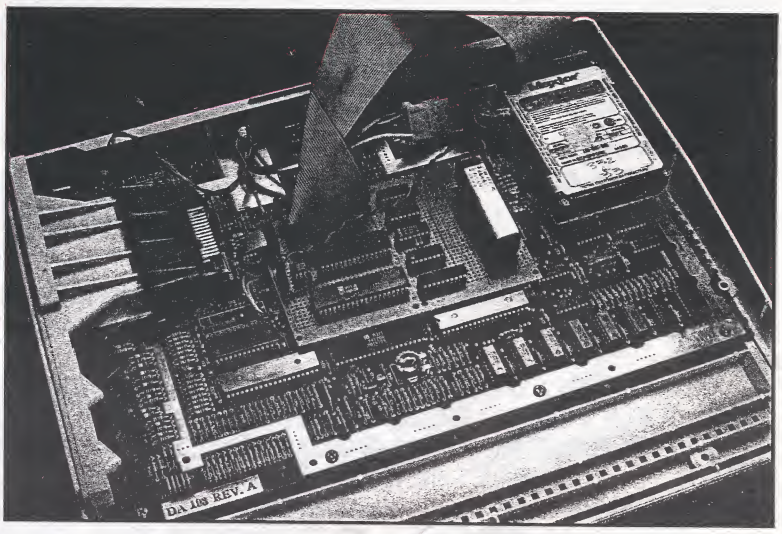
for the patient 8-bit user



## What Secrets Dwell Beneath This Innocent Ribbon Cable?

## It's Maxtor!

### 1200XL Built In Hard Drive!



## Article Inside!

## DEAN GARRAGHTY - FOR ATARI 8-BIT SUPPORT

NEW LOWER POWER PER POST PRICES! AVAILABLE FOR IMMEDIATE SHIPPING TO THE U.S. AT THE PRICES ARE SHOWN (IN US$):

QUICK Programming Language with Support Disks 1 AND 2: $26.00.

S.A.M. (Screen Aided Management) 80 column Desktop System. COMPLETE PACKAGE with Budget (Spreadsheet), Utils. 1, AND Designer DTP: $30.00.

Quick Ed. A Character Set Editor for Joystick, Mouse, or T/Tablet (DOES NOT NEED QUICK TO RUN): $8.00.

Glaggs It! (game): $6.00. Rubber Ball (game): $6.00. MineSweeper (game): $6.00. Bombi (game): $6.00.

*** PPP GAME PACK. All four above games for: $18.00 the set ***

## OUR OTHER PRODUCTS:

*Digi-Studio.* Digitized Sounds and Music Package: $12.00.
*Alien Blast.* A Space Invaders game for Light Gun only: $10.00.
*Demo Maker Pack.* Create simple demos with pictures, scroll lines, and music. Comes with Edit 7, a mode 7 graphics package, and an extra disk for using mode 8 and 15 pictures in your demos: $10.00.
*Maze Pack 1.* Contains Jawbreaker and Mousekattak from Sierra-on-Line: $15.00.
*Print-Filer.* This useful program is loaded before you boot your word processor, graphics package, etc. and sits in the background until you try to print from your chosen package. Now P-F springs into life and allows you to re-direct the print to a disk file. This file will then contain all commands that were to go the printer. This means that you can then print this file out from DOS, etc. without the original package being present. P-F also comes with an editor so you can join lots of files created from many different packages into one stand-alone file. You can mix text, graphics, etc. from any number of packages. Works with Print Shop, too: $10.00.
*TaskMaster.* This is a very powerful Tape to Disk transfer utility. Even has a dissassembler which allows you to fiddle with the code if you have an awkward tape you are trying to transfer: $10.00.
*News-Disk Pack 1-14.* All 14 issues of our old disk-based magazine. They all contain LOADS of articles and programs from the UK and other countries. Great reference material: $15.00 the set.
*News-Paper Issues 15/17/1819. Four* issues of our paper-based magazine (continued on from News-Disk): $10.00.
*News-Paper on Disk.* Issues 20-24 are currently available on disk with a printed cover. You print out each formatted page from the disk to create a full magazine: $6.00 the set.
*News-Paper Yearly Disk 1.* This disk contains extra bits and pieces we couldn't include in the News-Papers 15-19. It contains LOADS of QUICK source code and includes a full European-style demo written in QUICK with source code: $5.00.
*Rambit TurboLoad Kit.* This is a solder-in upgrade for XC12/1010 data recorders. Allows specially converted tapes to load SIX times faster than usual. Comes with conversion software on tape. Easy 15 min. installation: $27.00.
*Public Domain.* We currently have over 180 double sided PD disks available. We can offer AC readers the special price of just $2.00 per disk on 5 or more disks. See our catalogue for a full listing.

**ALL PRICES QUOTED ARE IN US$. PAYMENTS SHOULD BE MADE IN CASH.**
**A FULL CATALOGUE (INCLUDING A FULL LISTING OF ALL OUR PD DISKS) COSTS $5.00.**
**PLEASE ADD 10% (MINIMUM $5.00) FOR SHIPPING TO YOUR TOTAL ORDER VALUE.**
**PLEASE SEND BY REGISTERED MAIL IF SENDING A LOT OF CASH.**

DEAN GARRAGHTY, 62 THOMSON AVE, BALBY, DONCASTER, DN4 0NU, ENGLAND.
PHONE/FAX: +44-1302-855-026 (9AM to 6PM BST/GMT ONLY PLEASE!!)

# Atari Classics

Volume 5, Number 2
**Fall 1996**

EDITORIAL DIRECTOR
Bob Woolley
1161 Bay Street
Alameda, CA 94501
•RWoolley@ix.netcom.com
•CIS 75126,3446

ADVERTISING
Bob Scholar
2535 Tamalpais Avenue
El Cerrito, CA 94530
Subscribers may place noncommercial advertisements at no cost. These will be run in either a "Classified" section or in the "Letters" section.

HARDWARE & UTILITIES EDITOR
Charles Cole
1590 Foothills Drive
Sierra Vista, AZ 85635

EDUCATIONAL EDITOR
Bart Bresnik
555 Ware Street
Mansfield, MA 02048

GRAPHICS & ENTERTAINMENT EDITOR
Jeff Potter
814 Banbury Drive
Port Orange, FL 32119

Atari Classics is published no more than 6 times a year by Atari Classics, 1161 Bay Street, Alameda, CA 94501.

POSTMASTER: Send address changes to Atari Classics, 1161 Bay Street, Alameda, CA 945014.

SUBSCRIPTION PRICES: Six issues are $25.00 in USA, $30.00 in Canada, $32.00 elsewhere. Foreign subscribers may receive Air Mail delivery for an added fee.

BACK ISSUES, if available, may be purchased for 1/5 the price of a yearly subscription.

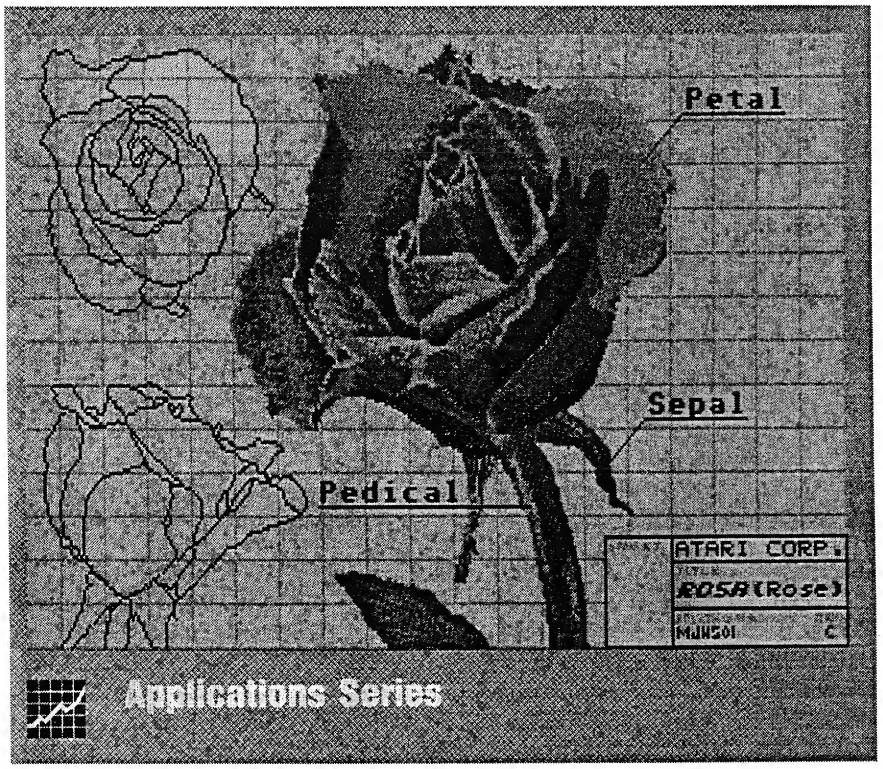
ARTICLES, REVIEWS or other material may be submitted for publication directly to the appropriate editor, to Atari Classics, 1161 Bay Street, Alameda, CA 94501or to rwoolley@ix.netcom.com.
Atari Classics takes no responsibility for return of submissions but will pay $25.00 for major feature articles upon publication.

# Jim Hood 1938-1996

Not long after agreeing to lead Atari Classics, Jim learned that he had developed a malignant tumor in his throat. He suffered through a major medical procedure that removed most of one ear and a portion of the left side of his neck, leaving him with impaired speaking and swallowing abilities. But, it did leave Jim, his family and his friends the possibility of a future. It allowed us to practice that vital self-deception that humanity has embraced - denial.

We lost. Jim is gone. He leaves a wife, Cindy and a grown son, Nathan. He also leaves those of us who knew him many memories of his artistry and wry humor. This AC issue is half Jim and half mine. His work is printed as he left it in remembrance.

He was a great friend and a wonderful human being. We miss him terribly.




Although the cover art on the NEOchrome packaging is credited to Jim Wood, it properly belongs to our friend Jim Hood. One of his many creations that added to our lives.

And, more sad news:

From: asharkis@aol.com (ASharkis)
Newsgroups: comp.sys.atari.8bit
Subject: ALEX PIGNATO
Date: 28 Jun 1996 11:59:21 -0400

The Ol' Hackers Atari Users Group is saddened by the loss of their founder and longtime president, Alex Pignato. Alex passed away suddenly on Wednesday, June 26.

Alex was a veteran of World War II, a retired trial attorney, a community leader in his town of Oceanside, NY and a true luminary in the 8-bit world.

He will be missed.

Alan Sharkis

# Readers' Writings

## Catching Up

Way back in Vol.3, #3 Lawrence Moon asked where he might purchase YEMACYB, a program to print color hard copy using a compatible black and white printer. Terry Johnson sent a prompt reply, which was misplaced and never printed. Here it is.

*Jim*

In response to Lawrence Moon's question – I bought **YEMACYB/4** from Innovative Concepts back in April, '92. Contact:

Mark Elliot
35 Wee Care Dr. Apt. #1
Mt. Clemens, MI 48043

...I'v got a question of my own. Is there anyway to send the "pipe" (!) from an Atari keyboard while using BobTerm? Any info would be greatly appreciated.

*Terry Johnson*

## Publish or Perish

I am writing to tell you the last issue I received was Jan./Feb. '96 – Vol. 5 - #1. It is now June, let me know what is happening.

Thank you,
*William R. Puente*

Since taking over publication of AC we have been unable to meet our schedule of six issues per year.

We started this year slow, but I didn't foresee any particular problems with getting six issues out. Bob Woolley did three articles in Volume 5 Number 1 and was working on some for Volume 5 Number 2, which took longer to complete than we anticipated.

We would have delayed his articles and used something else, except for the first time we had no backlog of submitted articles; so we had to choose between a very thin issue or letting publication slide again. We chose to wait. By the time Bob completed his articles we had also received a couple.

Your subscription is based on the number of issues mailed, so this has the effect of extending the time of your subscription.

If all this gets too frustrating for any subscribers, we will of course refund their unfilled subscriptions, but we hope that instead it encourages them to finally write that article they've been contemplating.

If you have an 8-bit article that you're looking to publish, but have better taste than to submit it to us, we understand that *Current Notes* may also be interested in submissions.

Articles for *Current Notes* can be submitted to Robert Boardman, Publisher, Hands On Communication, 559 Birchmount Rd. #2, Scarborough, ON Canada M1K 1P8.

Robert is 8-bit VP of the Toronto Atari Federation.

*Jim*

## More Memory

On the technical side I have a few questions for you Atari people in the know:

(1) I have an Atari 800XL with 256K expanded memory. Does anyone know of anyone else who ever increased the memory of an 800XL? Can it be done?

What is the limit? Could someone come along and place ONE MEG in an 800XL?

or more?

*Todd Sunrunner*

## International Users Group?

Now that I have been elected President of our local San Antonio Atari Group (AAAUA), I am more eager than ever to see if there is some sort of international association where ATARI users can all communicate and freely exchange ideas and questions with each other.

If there is no such international user group, there ought to be! I would be more than willing to begin a Newsletter to begin negotiations with all User Groups to create such an organization if one doesn't exist.

I do not know very much on the technical end of computers. I know just enough for me to run a small (Messages ONLY BBS) here in Texas. But I am Vice President of Marketing/Public Relations in my compny where I work and I DO KNOW about people connections!

I would be very happy if any Presidents of User Groups would contact me and just talk to me about this idea.

Thanks...
*Todd Sunrunner*
*ATARI Star Wars BBS*
*San Antonio, Texas*

## The SEGA Connection

Does anyone know if the Sega Joysticks and Joypads are compatible with the Atari Classic?

I haven't tried it [but] I read somewhere, long ago that they WERE compatible. Maybe someone could do an article on controllers available for the Atari Classic.

*James Martin*

## The SEGA Connection Continued

I'd like to make two comments concerning articles by Bob Woolley in AC Vol. 5, #1.

Thanks for the "Basic Skills–Soldering" story. It really helped when I had to desolder the OS out of my 65XE. I put in an UltraSpeed Plus OS, my second.

In "Controlling Servos", Bob talks about DB-9 connectors and their mounting tabs.

If you use joystick extension leads, either ATARI or SEGA, and a female DB-9 connector (on a small PCB), you don't need to trim off mounting tabs or cut up joystick cables. You won't have cables hanging from every gadget you ever put together. The only thing you need is two extension cables, which can be found "at your local SEGA dealer". And a female DB-9 connector for every gadget, of course. Put a male DB-9 connector on that PCB and you don't need the Y-connector (for the paddles) that Bob mentioned.

*Mathy G.F. Van Nisselroy*

## For Sale

Reconditioned disk drive sale.

All drives come with power supply, I/O cable, and 90 day warranty.

| | |
|---|---|
| 810 (no case) | $25 |
| 1050 | $35 |
| Rana 1000 | $40 |
| 1050 w/doubler | $45 |

Add $5/drive for S&H. A $15 credit will be applied if you send me your dead drive, power supply, and I/O cable. Make checks payable to:

*Paul V. Alhart*
*524 North Zee St.*
*Lompoc, Ca. 93436*

## Parallel Buss Interface? South Flordia 8-Bitters?

I am looking for [articles] that deal with the 800XL parallel buss interface.

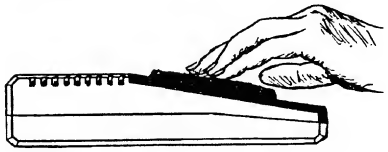
Are their any Atari 8-bit users groups or BBS's in south Florida? I am in Palm Beach County which is about 80 miles north of Miami.

*John Palhof*

I'm not sure that I understand your request, but we haven't printed any articles on the Atari 850, ICD P:R: Connection or other interface that has a parallel connection.

The only Florida user group that we list is the Pinellas Atari Computer Enthusiasts in Largo, FL. The last time I looked, they had an entertaining newsletter. It had a hand decorated Christmas cover a few years back that is still one of my favorites.

*Jim*



# Editorial

## Where Do We Go From Here?

Jim Hood was more than just a friend of mine - he was also a major force in the look and presentation of Atari Classics. This magazine, however, is written by users who continue to benefit from their Atari 8-bits. As long as we have users, we'll have AC. It won't look as professional. There will be little or no editing as such. Many of the little touches of humor will be absent ("for the dedicated 8-bit user" =>"for the patient 8-bit user"). Jim will just not be in AC anymore. But, the rest of us will.

### People

I have assumed all of the responsibilities of Atari Classics as of January 1, 1997. This includes any outstanding liabilities - all subscriptions will continue in force until term. For those of you who are familiar with my tendency to bury correspondence, my wife, Sharon has agreed to assist me in the administration of the magazine. Bob Scholar will continue to duplicate our disks, which will ship with every other issue. I have not been in contact with the most important group however, those that submit articles and advertising.

We have not made it easy for an enterprise to advertise in AC, I'm afraid. Ad copy submitted in May that is printed in October is going to be somewhat out of date..... It tends to cause difficulties between customers and dealers when the items advertised in this new Atari Classics issue have been sold out for months. And, public response to an ad is difficult to measure when the ad appears six months after it was scheduled. Help us out a little. If you do see something in AC that interests you and you contact the vendor, tell them where you heard about it, OK? We'll try to do a little better on this one, too.

A number of folks have expressed a willingness to help out. My problem is that I don't have a clear idea of how to utilize these resources. So, I'm going to generalize about what AC would like to see. Articles about some aspect of Atari 8-bit computing based on your own experiences in solving a problem, developing a process or information about your computer. New or existing sources of hardware, commercial or public domain software, and printed material. BBS and Internet sites. News items. These do not need to be in the form of a complete article - be our eyes and ears. Thanks!

### Places

Our new address is:

Atari Classics
1161 Bay St.
Alameda, CA 94501
CIS - 75126,3446
rwoolley@pacbell.net

Remember - if you do not get a reply in a reasonable length of time, send a follow-up addressed to "Sharon" for a prompt response.

### Things

The magazine itself will undergo a few production changes. The cover(s) will be printed on metal plates in a press as always. The inner pages will be photocopied, however. The nature of the text and graphics in the articles reproduces well on a Xerox while the detail and coverage on the covers suffers visibly. So, we can save a lot of time and labor by splitting the output. The covers and ads can be produced weeks before shipping and the texts can be done the day before if necessary.

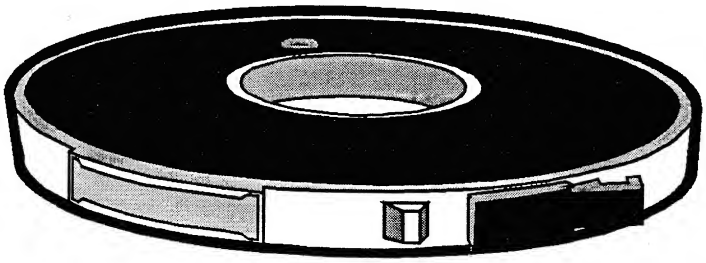
We will continue to produce about 20 pages of text. The additional pages (ads, etc.) will now normally be 12 or less, giving us 28 to 32 pages per issue. It may seem thinner, but the same amount of meat is there - just less bread in the "sandwich".

I do have Jim's HD from his TT, which contains all the AC files and graphics. Unfortunately, Jim used an Atari and I use an IBM for DTP. My 1040ST can read the files OK, but many of the applications that were running on Jim's 32MB TT seem to choke on my 1MB machine. I got some of the graphics and, most importantly, the subscriber files. I'll keep the drive just in case. Someone should be able to pry those files out of there if it turns out we need them!

I guess I'd better get this guy in the mail.... write to me and tell me what you think. Thanks for listening!

**Bob Woolley - Atari Classics  February 1997**

# Atari Classics Disk - Volume 5&6



## A Word From Our Sponsor

This issue contains a disk for those of you that requested ($) them. In the past, the disks have been a combination of 8-bit introductions from Nir Dary, program listings from the last two issues, and more PD whatevers from Nir. I liked them and I hope most of you enjoyed them, also.

Due to circumstances beyond our control, AC will have to forego the added features of the issue disks and just supply the program listings for the two current issues. The disks will be SD and SS, cutting our duplication time in half. We will use DOS 2.0 formatted disks and include DOS.SYS and DUP.SYS on each disk. Other than that, we are going to play this by ear. Let us know how you feel about this - we have limited resources and we are willing to apply them where *you* want them.

Atari Classics

## On The Disk

From the last issue, we have the SERVO.BAS and SERVO.ASM files. You may notice that the SERVO.BAS file does not contain all of the REMark entries that were listed in the article - just the code.

From this issue, we have Paul Alhart's FORMAT4.ASM and FORMAT4D.ASM files. These were actually MAC65 files, but you can ENTER them into EDASM. You cannot, however, assemble them successfully because of the way other assemblers handle memory addresses of labels. Line 150, for example, reads:

```
0150  LDA # <DEVICE
```

which loads the low order byte of the address for the data at location DEVICE. Not the data, but the address of the data, OK? You must change this in EDASM to:

```
0150  LDA #DEVICE & $00FF
```

which accomplishes the same result in EDASM. Likewise, line 170:

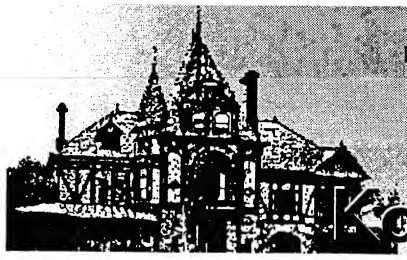
```
0170  LDA  # >DEVICE
```

which loads the high order address byte of label DEVICE, must be:

```
0170  LDA #DEVICE/256
```

Now, you can assemble just fine.

The SmartIDE code, MOVEROM2.ASM (which is no different than MVROM.ASM from SmartOS) and MOVERAM2.ASM are included as well as the IDE driver code itself, IDE38.ASM. These are relatively straightforward EDASM files which should assemble as is.



# The Garret

## Komputrz & Kre8ivity

Ed Hall

Ray Bradbury, whose *Martian Chronicles* was recently converted to a CD-ROM adventure, was asked by People magazine (11/27/95) if he owned a computer. Ironically, his response was:

"No. I write on an IBM Wheelwriter. I could out-type you any day. I write a short story in three hours, and I don't change it. Computers are for changes."

Personally, I'm not sure I'd want to brag about never revising a story—for one thing it makes writing seem a bit too oracular. A more balanced view comes from David Cronenberg (*Scanners*, *The Fly*) and Salman Rushdie (*The Satanic Verses*, *The Moor's Last Sigh*). In the July/August 95 issue of *Shift*, a magazine on "media culture," the following exchange appeared.

**Cronenberg:** Now J. G. Ballard [whose book Crash Cronenberg has just adapted in his latest screenplay] being the prophet of technology that he is, said he can tell when a novel has been written on a computer.

**Rushdie:** I think that's crap.

**Cronenberg:** I do too. He said, "They just go on and on." And I said, "You know, people have written by hand and gone on and on."

**Rusdie:** In my view, my writing has got tighter and more concise because I no longer have to perform the mechanical act of re-typing endlessly. And all the time that was taken up by that mechanical act is freed to think. So I have more thinking and less machine time.

**Cronenberg:** I remember almost not changing a sentence that was bad because it would mean cutting and pasting.

**Rushie:** Of course. And I had this kind of fetish about presenting clean copy. I don't like presenting my publisher with pages with lots of crossings-out and scribbling. So I would be manic at the end of typing a page where actually I didn't want to change anything, not at all... So there's no doubt in my mind that the computer's improved my writing. And for exactly the opposite reason of what Ballard says.

### Plots Unlimited

"Confused by all the computer programs for writers?" asks an ad in *Writer's Digest*. Quite frankly, yes, though not for the same reason the ad suspects. Way back in **AC**'s first issue, I mentioned Plots Unlimited, a piece of software for generating stories on PCs and Macs. I thought it was a pretty goofy idea. Well, an intriguing goofy idea. I mean, if a computer program can defeat the reigning world chess champion at least once, why can't it write a good story? Or, what is perhaps more reasonable, write something highly formulaic, like a sitcom?

Apparently there's lots of authorware in the Mac/PC world, everything from programs like ScriptWare, which sets up the specialized formats required for film and TV scripts, to programs that do the actual writing for you. SportsWriter, for example, is a program in use today by many small-circulation newspapers. Punch in data from a high-school basketball game and the program spits out "fairly generic" copy. There's even a wp that completes words before you finish keying them in.

Where will it all end? I am reminded of that old, Nebula-award winning story by Robert Silverberg, "Good News from the Vatican," in which a computer becomes pope.

As you can tell, the whole business fascinates me, and when I saw the latest incarnation of Plots Unlimited, I just had to get it! Why? Because (hold on to your joysticks!) P.U. is now available to all Atari users, both 8- and 16-bitters. Yep, it's true. In fact, Plots Unlimited is available in a universal format which any computer owner can use. It's a book.

So I ordered the dang thing. Remember "paint-by-numbers"? Well, P.U. is write-by-numbers, a database of plot elements which you use in much the same way as those "choose-your-own adventure" books. The plot elements are stated in very general terms—"Jack's wife, Carol, desperately needs money for a certain purpose." You can quite quickly cobble together a plot, but you will have to supply the details and the dialogue to give the story its spark.

A writing aid like P.U. is actually quite logical when you stop to think about it. With the dominance of TV as an art form (well, I use the term loosely), and with the resulting decrease in the attention span of viewers, plot is assuming a greater relevance in fiction. Stories zip along at breakneck velocity—so fast, in fact, I am certain many of today's films would be incomprehensible to people not reared on them. Their speed (measured in "jolts per minute") is a function of plot, never character or setting. In fact, for many hit TV shows a single plotline is not enough; usually there are two of them going at the same time.

Writers of "serious" literature may scoff at P.U. as a producer of fodder for plot-hungry markets like sitcoms and thrillers, but so too did chess-players once scoff at the idea of a computer beating an international grand master. Deep Blue, the IBM supercomputer which managed to steal a game from Kasparov, considers more than 100 million moves per second. What if IBM were to build Deep Plot, a computer devoted exclusively to story construction?

## Microserfs

From a computer program that can help write a novel, we go to a novel about computer programmers who need help. However, *Microserfs* by Douglas Coupland (originator of the term "generation x") is not the type of novel that could have been generated by P.U. because it's nearly plotless. The story, such as it is, revolves around a group of programmers working for Microsoft in Seattle. They quit and move to Silicon Valley to start up their own software company. That's the plot.

What makes the book so much fun is the sheer brilliance of the writing. The novel is mostly a long series of discussions and animadversions on life in the '90s, a search for meaning and identity by twenty-somethings whose main connection with reality ("Daniel, do I seem alive?") is consumer products ("...we looked for an Italian restaurant so we could re-enact the classic Lady and the Tramp spaghetti-noodle/kiss scene."). *Microserfs* is a novelized Trivial Pursuit, an intellectual McDonald's, a combination of Wayne's World and Plato's dialogues for '90s consumers.

Daniel and his friends are lost souls, computer geeks out of sync with life, themselves, everything. Even when they focus on their bodies, they cannot avoid overdoing it: "Apparently after you body-build, you have a two-hour time window in which your body can suck up amino acids. This is your protein window. I was talking to him and he said, 'Man, I'd like to talk some more, but my protein window is closing,' and he ran off to the kitchen and ate a chicken."

Daniel faints a lot, Susan has problems with her "dating architecture," Karla is a recovering anorexic, Bug discovers he's a homosexual, Michael falls in love with someone named BarCode on the Internet, and so on. Yet because they're all brainy programmers, Coupland sees them as contemporary high priests who examine not entrails but consumer products: breakfast cereals, Gap clothing, tampons, Star Trek, Barbies, etc. Everything has a subtext.

Their penchant for introspection is evident even in the name they choose for their company: Interiority. Daniel starts keeping "subconscious files" on his computer, which are reproduced throughout the novel like illustrations, whole pages taken up by lists of words in fonts of varying size. Other pages are filled with a single word, 0s and 1s, consonants, vowels, or encrypted words:

"She sed th@t we, az humanz, bear the brdn uv havng 2 B evry animl in the wrld rold in2 1. She sed th@t we rEly hav no identiT uv our own. She sed, 'Wh@ iz human B-havior, X-ept tryng 2 proov th@t w'r not animalz?' She sed, 'I think we hav strAd so far awA from our animal originz th@t we R bent on kre8ng a noo, soopra-animal idNtiT.' She sed, 'Wh@t R komputrz but the EvryAnimalMashEn?' I kouldn't BlEv she wuz talkng like this. She wuz like an episode of Star Trek made flesh."

There are the usual high-jinks associated with computer companies. At Microsoft "we lapsed into one of our weekly-ish communal stress-relieving frenzies—we swiped sheets of bubble-pak from the supply rooms and rolled over them with our office chairs, popping hundreds of plastic zits at a go. We punished plastic troll dolls with 5-irons, blasting them down the hallway, putting yet more divots in the particle board walls and the ceiling panels. We drank Tabs and idly slagged interactic CD technology (Todd: 'I used the Philips CDI system—it's like trying to read a coffee table book with all of the pages glued together.')."

The book is a hoot for 300 pages, then suddenly becomes a bit tiresome, a bit self-indulgent, as Coupland maneuvers us toward the ending. Though he has cleverly prepared us for the finale by means of the "subconscious files," it takes too long in coming, and the conceit, while interesting, isn't strong enough to carry the weight of a satisfying conclusion.

Of course, endings are notoriously difficult to write. (Hm, is this a product niche which needs filling?) In the meantime, read and enjoy the book, and if at any time you lose patience with it, skip ahead to page 360 to see how it turns out. Then put it in a time capsule; it'll have great nostalgia value. As Daniel says, "The great Atari gaming collapse of 1982 (*sigh* I remember it well)."

## VARIABLE Opera

Finally, I offer you my own bent example of how computers can inspire creativity—a short story written in the form of a BASIC program. (Er, please don't type it in.)

## OOPS!

There was a mistake in the chess problem solution printed in Vol.5, #1. Black's first move should be P-Q4 not P-QB4. Thus:

| | | |
|---|---|---|
| 1 | N-KB3 | P-Q4 |
| 2 | N-K5 | N-KB3 |
| 3 | N-B6 | KN-Q2 |
| 4 | NxN | NxN |

Knew we shouldn't have consulted that Deep Blue programmer.

```
-------------
CHARACTER SET
-------------
100 Bad Irving - he refuses to clean up after
his horse.
120 Sheriff Bob - he's replacing hitching posts
with parking meters.
130 Wilbur - he drinks from the horse trough.
140 The Parson - the parson.
150 Miss Daisy - a right purdy gal.
160 Diamond Lil - it's rumored her children are
legitimate.
170 The Schoolmarm - she's got a lot to learn.
180 Johnny Muskrat - some call him Chief.
190 Squeegee - they found him in the desert.
200 Professor Periscope - he's building a
strange contraption.
210 Also appearing - the cardsharp, the
greenhorn, the town drunk, the old prospector,
the mysterious drifter, the Sheriff's horse.
```

```
--------
DIALOGUE
--------
220 Ghost Gulch ain't big enuff fer the two of
us.
230 Looks like they figgered out my true
identity.
240 I love you.
250 I hate you.
260 Which orifice shall I use?
270 Set your lasers to fry.
280 We'll head 'em off at the past.

-------------
SOUND EFFECTS
-------------
290 Blam!
300 Thud!
310 Zzzzzt!
320 Vrooom!
330 Squeakedy squeakedy squeakedy…

---------
PLOT DATA
---------
340 Sheriff Bob asks Bad Irving to be his
partner.
350 Wilbur stumbles across the Professor's
contraption.
360 The parson receives a mysterious phone call.
370 Miss Daisy straps on her shootin' irons.
380 Diamond Lil gives the greenhorn a lesson in
bareback riding.
390 Johnny Muskrat comes down with smallpox.
400 Squeegee falls in love with the Sheriff's
horse.
410 Professor Periscope's evil twin shows up.
420 The cardsharp and the drifter visit the
souvenir shop.
430 Wilbur gets an erection.
440 The schoolmarm uses her magical powers.
450 The old prospector is possessed by a being
from another dimension.
460 The town drunk wakes up in the Sheriff's
dungeon.
470 Bad Irving blows up the galaxy.

-------
SETTING
-------
480 Ghost Gulch is a town, dude ranch, theme
park, BBS, planet, game, metaphor.

--------------
ERROR MESSAGES
--------------
490 Out of ammo.
500 Nonexistent device.
510 Stack overflow.
520 Wrong dimension.
530 Character flaw.
540 Plot malfunction.

--------------
TEMPORAL ARRAY
--------------
550 19th century
560 20th century
570 21st century.

---------
VARIABLES
---------
580 Horse, Soap, Space
590 Shootin' iron, Uzi, blaster.
600 Hitching post, parking meter, parking orbit.
610 Bootleg still, cold fusion gimmick, time
machine.
620 Sysop, dungeonmaster, narrator.
630 God, prime mover, programmer.
640 Conflict, climax, resolution.

---------
CONSTANTS
---------
650 Death and taxes.
660 The speed of light.
670 Software-driven spacetime.

---------
MAIN LOOP
---------
680 Someone falls in love.
690 Someone gets shot.
700 Someone goes away.
710 A beautiful sunset.
```

# 1200XL CleanUp

**by Bob Woolley**

For those of you that are following the hardware modifications series in AC, let me recommend my favorite raw material, a 1200XL. These 'orphan' versions of the Atari 8-bit were probably the reason that the computer division ultimately failed to capture a significant market. Overpriced, incompatible, and lacking any add-on capabilities, the 1200XL was a dismal marketing effort. BUT, we aren't interested in it's failures, only it's strengths. It was supposed to be a $700 personal computer. As such, it has a very nice keyboard, fully socketed ICs, four function keys that are programmable and pleasing packaging. Packaging that has plenty of room inside for all sorts of goodies and sockets to plug things into..... Now, as they come from Atari, the 1200XL is a bust. The video to a monitor just plain does not work, the SIO is not powered and there is no PBI port. Which is why many of these jewels are offered for sale at very low prices by users who prefer a less cranky system than a 1200XL. Of course, if you are willing to spend some time on them, you can do very well. This article will tell you how.

You will need to acquire two ICs from an XL/XE computer - the OS ROM (CO61598B) and the PAL (CO61618). I got mine from an old 600XL, one of the better supplies you've probably never heard of. These 600XLs were almost worthless as computers, but they were ALL socketed, come with a decent 800XL keyboard and power supply, and cost almost nothing. They are a great source of parts and often are available for less than $10. However, collect your parts and warm up the old iron.....

The following modifications will be made to the 1200XL:

* Add power to the SIO connector. This will allow you to use the MPP1150, a P:R Connection, an XM301, (oh, boy!) or any other SIO device that is powered by the SIO buss itself.
* Clean up the video output. This is a minimal change that will give you a decent signal. It is not the optimum solution - see the video issue of AC if you want really killer output.
* Install the OS from the 800XL/130XE systems. This also allows you to use a single chip custom OS such as OmniMon or BossXL.

Your first task is to open up the case. This should not be a problem for you. If it is, I would reconsider taking on this or any other of my projects. Once inside, remove the shields and toss them in the trash. That last activity precludes you from using the 1200XL on a television thru the tuner. Get a nice monitor or a TV with A/V inputs. Or, both!

These modifications are essentially the removal of some of the factory components and replacement of others. To do this, you must become familiar with the ICs and other major parts in the 1200XL. Take some time and review where and what each IC is on the top of the board and their location when you are looking at the underside. The part designator is just above each IC with the exception of U27 where you'll see it on the left of the chip. The top of the PC board is called the component side and the bottom is called the wiring side (even though we have traces or wiring on both sides). OK? Good - I have built a list of the parts that need to be altered for you to follow. The left hand column is the part designator on the PC board. The second column is a description of the part. The third column is an approximate board location. The right column is what replaces the part - wire, another part or nothing (---). If you look closely at the board on our cover, you may notice that the 'wire' that I specify looks like a resistor. It is, actually. I use 1 ohm resistors instead of an actual wire. I just like the look of them better - you can use either.

In some cases, there is nothing inserted in a component location - W13, for example. Just clear the solder from the hole and add the new part. It is easiest to remove all the specified parts first and then add in the new components. Check them off as you go along.

After you have finished changing components, take a look at the U12 and U13 sockets. You may have either 24 or 28 pin sockets in those locations. If there are 28 pin sockets installed, you should also have the proper W jumpers (W11, W12, W13 and NOT W7, W8, W9). If you have 28 pin sockets, you can just use them as they came from the factory. If you have 24 pin sockets, you must remove them and install a 28 pin socket in U13. In either case, the result must be a 28 pin socket in U13, jumpers in W11-W13 and no jumpers in W6-W9.

The last wiring to be done is the addition of a wire on the bottom of the board from the 6502 (U21), pin 23 to the top hole of W6, which was removed. This connects the A13 address line to pin 26 of the OS ROMs.

Remove the PAL at U14 and plug in the XL PAL (CO606818). Plug the XL OS (CO606818) into U13. Now, assemble the 1200XL and try it out. Should look and run just like an 800XL, except you now have the real function keys and a very nice keyboard. And room for the SmartOS.... And an IDE drive.....

And ????

| Name | Description | Location | Replace With |
|---|---|---|---|
| R63 | 100 ohm resistor | below SIO conn | wire |
| L15 | 100uh inductor | above U27 | wire |
| C115 | 10uf capacitor | above U27 | wire |
| CR20 | signal diode | above U27 | --- |
| R21 | 2.2K resistor | above U27 | 3.9k resistor |
| W6 | 0 ohm resistor | right of U13 | --- |
| W7 | 0 ohm resistor | above U12 | --- |
| W8 | 0 ohm resistor | above U12 | --- |
| W9 | 0 ohm resistor | above U12 | --- |
| W11 | --- | above U12 | wire |
| W12 | --- | above U12 | wire |
| W13 | --- | above U12 | wire |

# A—T—A—R—I
# Answers, Tips And Relevant Information

## Programming Your Disk Drives

Paul V. Alhart

Although I will be using Assembly language programs to demonstrate some of the concepts in this column, don't be scared off. The programs will be available "ready to run" on the Atari Classics disk. The listings are commented and hopefully easy to follow. Besides, you just might learn something if you stick with me.

Over the years since I purchased my first Atari, I have invested a considerable number of hours of learning/programming time to save myself a few seconds here and there. If you understand that (even better if you can relate to it) we are ready to proceed. Several of those well spent hours will now be discussed.

It all started shortly after installing the US+ O/S upgrade in my 800XL. US+ automatically took care of setting up the RAMdisk handler at power up, but I still had to go to DOS to format it. With some software this was quite inconvenient if not impossible, not to mention being a waste of my time. Between the *Atari Personal Computer System Operating System User's Manual* and Compute's book *Mapping the Atari*, I was able to write the short AUTORUN program shown in listing 1, that would format my RAMdisk (disk #4) at boot-up.

I immediately added this program to my BobTerm disk so all my downloads could be saved directly to my RAMdisk. It wasn't long till I ran into good old ERROR 162 (DISK FULL). OK I thought, I'll just rewrite the above program to Format my RAMdisk in double density. Here is where hours turned to days. I didn't know how to reconfigure a drive without going back to DOS and none of my books hinted at the answer. I did the only thing I could. I logged on to my favorite FIDO NET BBS, went straight to the Atari 8-bit Echo, and said "HELP". My FIDO friends came through with flying colors. I learned that the MyDos documentation had most of the information I needed. Also, Bill Wilkinson had written an *Insight: Atari* column on this topic in the October 1985 issue of *Compute Magazine*. One kind soul even mailed me a copy of Bill's article. With this information in hand it was back to my learn/try/learn some more method of programing. Here is what I learned.

### The Percom Configuration Block

Back in the early days Percom Data Corp. wanted to market both single and double sided drives using both single and double density to the Atari market. They also wanted them to be compatible with the Atari 810 and (soon to be released) 815 disk drives. To do this they developed the Percom configuration block. This set the standard followed by all third party drive manufactures. The only exception being Atari's own 1050 drive which operated in single and enhanced density. Yech!

NOTE: A Happy 1050 or US Doubled 1050 does use the Percom configuration block.

This configuration block consists of 12 bytes of memory within the drives control microprocessor. You can read a drive's configuration block by passing an "N" to it as an SIO command. You can write a new configuration block with an "O" command. This is similar to the "R" "W" commands for sector input/output except the data length is always 12 bytes and no sector number is needed. Also note that all double byte values are in high byte/low byte order. The 12 bytes and their usage are:

| Byte # | Use |
|---|---|
| 0 | Number of tracks (40 for a standard drive) |
| 1 | Head step rate (setting varies by manufacturer) |
| 2 – 3 | Sectors per track (0 and 18 for a standard drive) |
| 4 | Number of sides (0 = single sided; 1 = double sided) |
| 5 | Density (0 = single density; 1 = double density) |
| 6 – 7 | Bytes per sector (0 and 128 for single density) |
| 8 | Translation Control<br>bit 7: 1 = 40 trk. disk I/O on an 80 trk. drive<br>bit 6: Always 1 to indicate drive present<br>bit 1: 1 = Handle sectors 1-3 as full size sectors<br>bit 0: 1 = Sectors number 0-17 (for example), not 1-18 |
| 9 | Serial rate control (setting varies by manufacturer) |
| 10–11 | Miscellaneous (setting varies by manufacturer) |

Since some of the 12 bytes are not used or mean different things to different drives, it is best to first read the configuration block of the drive you want to configure. Then plug in new values to just the appropriate bytes to make the changes you need. After reconfiguring a drive, DOS needs to be updated with the changes. This is easily accomplished by doing a JSR to DINIT. One problem here is that DINIT is located in the File Management System portion of DOS, and not all DOS's use the same memory location for DINIT. My example program FORMAT4D, in listing 2, uses the DINIT location used by SmartDOS, DOS XL, and AtariDOS 2.0. To use this program with any other DOS you must first determine the location of DINIT and then change it in the EQUATES portion of the program.

### Putting It All Together

FORMAT4D (listing 2) uses all the information we have learned so far. It first reads the drives current configuration. If it is configured as single density (the default at power up) everything proceeds normally. If it is already configured to double density, the program terminates with no further action taken. This allows me to re-boot without wiping out any data that may be stored on the RAMdisk.

Next the screen border is set changed to red to indicate the drive will be reconfigured and formatted in double density.

The drive is then reconfigured for double density, DOS is informed via DINIT, and the drive is formatted.

With this information and some neat utilities like RAMcopy and Multi Autorun System you can have your RAMdisk configured, formatted, and loaded up with support files. All done automatically at power on. See what I mean? A few weeks of learning/programming can save you precious seconds down the line.

## CREDITS

**Atari Disk Drive Compatibility** by: Bill Wilkinson. Insight: Atari, Compute Magazine, October '85.

**RAMcopy** by: Charles F. Johnson. Analog, July '86. Lets you automatically copy files to your RAMdisk at power up.

**Multi-Autorun** by: Bill Bodenstein. Antic Magazine, March '87. Modifies Atari DOS 2.0 or DOS 2.5 to allow up to 27 AUTORUN files to be executed during boot up.

**Ultra Speed Plus O/S** by: Robert Puff, Computer Software Services, Rochester, N.Y. (716) 429-5639. This is 3 switchable operating systems in one package. Sets up RAMdisk handlers, reads/writes/formats in Ultra Speed Skew on drives that support U/S, and much-much more.

**BobTerm** by: Robert Puff. May be the best terminal program ever for the Atari.

**MYDos** by Robert Puff. Excellent third party DOS for your Atari. Includes support for hard drives, double density, RAM drives, sub-directories, and lots more.

**FIDO NET**: A world wide net of private run BBSs. A great free source of information, help, and used equipment. More on FIDO in the next A-T-A-R-I.

Listing 1

```
10 ;FORMAT4
20 ;
30 ;BY PAUL ALHART
40 ;
50 ;
60 ICCOM =   $0342
70 ICBAL =   $0344
80 ICBAH =   $0345
90 CIOV  =   $E456
0100 ;
0110     *=  $5000
0120     LDX #$20
0130     LDA #$FE    ;FORMAT COMMAND
0140     STA ICCOM,X
0150     LDA # <DEVICE
0160     STA ICBAL,X
0170     LDA # >DEVICE
0180     STA ICBAH,X
0190     JSR CIOV    ;DO IT
0200     RTS
0210 DEVICE .BYTE $44,$34,$3A,$9B ; D4:
0220     *=  $02E0
0230     .WORD $5000
```

Listing 2

```
10 ;ALHART ENTERPRISES PRESENTS
20 ;
30 ;        FORMAT4D
40 ;      AEP  3/27/93
50 ;
60 ;FORMAT4D is an AUTORUN utility.
70 ;It reconfigures Drive #4 to Double Den-
    sity,
80 ;informs DOS of the change, and then For-
    mats Drive #4.
90 ;
0100 ;To use with any DOS other than
0110 ;ATARI DOS 2.0 -DOS XL -SMARTDOS
0120 ;DINIT may have to be changed.
0130 ;
0140 ;
0150 DDEVIC = $0300
0160 DUNIT = $0301
0170 DCOMND = $0302
0180 DSTATS = $0303
0190 DBUFLO = $0304
0200 DBUFHI = $0305
0210 DTIMLO = $0306
0220 DTIMHI = $0307
0230 DBYTLO = $0308
0240 DBYTHI = $0309
0250 ICCOM = $0342
0260 ICBAL = $0344
0270 ICBAH = $0345
0280 DINIT = $07E0
```

```
0290 DSKINV = $E453
0300 CIOV =  $E456
0310 ;
0320      *=  $5000
0330 ;
0340 ;CHECK IF DRIVE 4 IS ALREADY DOUBLE
DENSITY. IF YES THEN END.
0350 ;
0360      LDA #$31     ;DEVICE TYPE
0370      STA DDEVIC
0380      STA DTIMLO
0390      LDA #$04     :DRIVE #
0400      STA DUNIT
0410      LDA #$4E     ;GET CONFIGURATION
0420      STA DCOMND
0430      LDA #$40     ;READ CONFIGURATION
                BLOCK & STORE IN TBLOCK
0440      STA DSTATS
0450      LDA # <TBLOCK
0460      STA DBUFLO
0470      LDA # >TBLOCK
0480      STA DBUFHI
0490      LDA #$00
0500      STA DTIMHI
0510      STA DBYTHI
0520      LDA #$0C     ;12 BYTES
0530      STA DBYTLO
0540      JSR DSKINV
0550      LDA TBLOCK+5
0560      CMP #$04     ;SEE IF DRIVE IS CON-
                FIGURED FOR DOUBLE DENSITY
0570      BEQ RTS
0580 ;
0590      LDA #56      ;COLOR BORDER RED
0600 ;
0610      STA 53274
0620 ;
0630 ;SET D4: FOR DOUBLE DENSITY USING PER
COM CONFIGURATION BLOCK.
0640      LDA #$31     ;DEVICE TYPE
0650      STA DDEVIC
0660      STA DTIMLO
0670      LDA #$04     ;DRIVE #
0680      STA DUNIT
0690      LDA #$4F     ;$4F = RECONFIGURE
0700      STA DCOMND   ;$4E = GET CONFIGURA-
                TION
0710      LDA #$80     ;$80 =WRITE NEW CON-
                FIGURATION THAT IS STORED
                IN CBLOCK
0720      STA DSTATS   ;$40 = READ
0730      LDA # <CBLOCK ;CBLOCK ADDRESS
0740      STA DBUFLO
0750      LDA # >CBLOCK
0760      STA DBUFHI
0770      LDA #$00
0780      STA DTIMHI
0790      STA DBYTHI
0800      LDA #$0C     ;12 BYTES
0810      STA DBYTLO   ;IN CBLOCK
0820      JSR DSKINV
0830 ;
0840 ;LET DOS KNOW DRIVE CONFIGURATION
BEFORE FORMATTING.
0850      JSR DINIT
0860 ;
0870 ;FORMAT DRIVE #4
0880      LDX #$20
0890      LDA #$FE     ;FORMAT
0900      STA ICCOM,X
0910      LDA # <DEVICE ;DEVICE ADDRESS
0920      STA ICBAL,X ;DEVICE
0930      LDA # >DEVICE
0940      STA ICBAH,X
0950      JSR CIOV
0960 RTS RTS
0970 ;
0980 CBLOCK
0990 ;TWO BYTE NUMBERS IN CBLOCK
1000 ;ARE IN HIGH BYTE-LOW BYTE FORMAT.
1010      .BYTE 40     ;# OF TRACKS
1020      .BYTE 1      ;STEP RATE
1030      .BYTE 0,18   ;SECTORS/TRACK
1040      .BYTE 0      ;# OF HEADS-1
1050      .BYTE 4      ;DENSITY 4=DD 0=SD
1060      .BYTE 1,0    ;BYTES/SECTOR
1070      .BYTE 255    ;DRIVE SELECTED? Y/N
1080      .BYTE 0      ;SERIAL RATE CONTROL
1090      .BYTE 0,0    ;NOT USED
1100 ;
1110 TBLOCK
1120      .BYTE 0,0,0,0,0,0,0,0,0,0,0,0
1130 ;
1140 DEVICE
1150      .BYTE $44,$34,$3A,$9B ;D4:
1160 ;
1170      *=  $02E0    ;RUN AT $5000
1180      .WORD $5000
1190      .END
```

# Dallas "ROMs"

by Bob Woolley

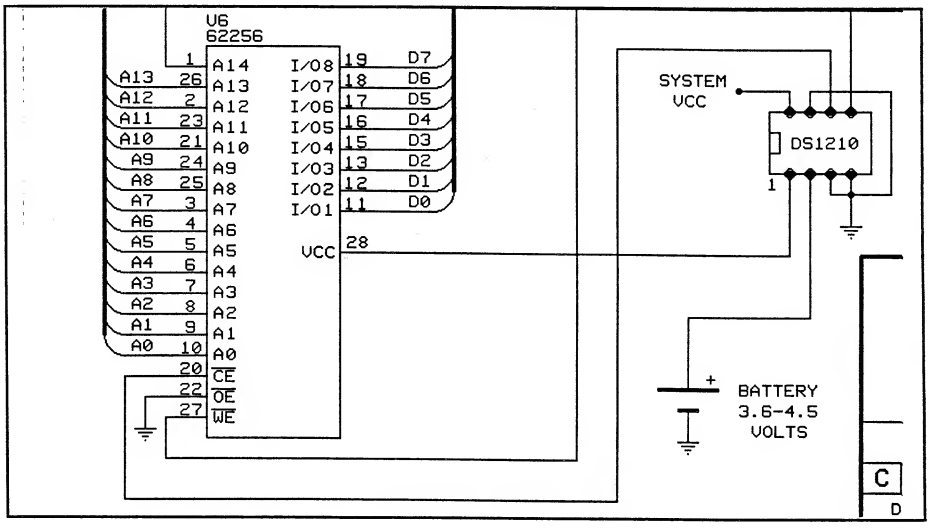
In the SmartOS modification that we had last year, the core of the project was a non - volatile static RAM. The stock Atari is built with either an EPROM or ROM for the storage of the operating system code. These devices retain whatever data is written to them without the aid of external circuitry or power. The problem is, you cannot alter their contents once they are programmed. Any changes you would like to make in the OS require the removal and replacement of the chip. The cartridge, MIO interfaces, the 1050 and 850, also have ROMs or EPROMs that store code. Changing code in any of these devices is difficult. The static RAM devices, such as the 62256, have the same pinouts as an EPROM of equal size and can be altered just like any RAM memory. Unfortunately, the data in a RAM will not survive a power down, Once the power is turned off, all data is lost. The SmartOS presented a method of retaining data in one of these RAM chips through numerous power cycles, giving us the ability to alter our OS as often as we like.

Never being content to leave something alone, I have made a few modifications to the SmartOS modification. The alteration I will describe here is to the method used to retain data in our static RAM chip. In the first design, I used a Dallas Smart Socket to provide all the necessary components. All you did was plug in a static RAM and you were up and running. This makes for a simple project but it left something to be desired from my point of view. First, the battery was internal and not replaceable. Sure, they last 10 years. Usually. I can just imagine my RAM chip blowing it's brains out and shorting out the battery in the Smart Socket. No more Smart Socket. I can also see wanting to use a larger RAM (128K or 512K). Can't use a single Smart Socket for that either... Or multiple RAMs. What was needed was a little more flexible setup. And, if it's a few bucks cheaper - great!

Take a look at the Smart IDE schematic. The lower right hand corner is a sketch of the data retention circuit. Again, we use a Dallas Semiconductor part, but instead of the whole assembly, we only use the controller chip - a DS1210. This chip provides the sensing and control circuits to prevent data loss while power is being cycled, completing any write cycles and switching in the battery when needed. The battery may be either a PC style 3.6 volt lithium cell or three ordinary penlight batteries (4.5 volts). Either battery will supply much more power than is available in the Smart Socket and can be replaced easily and cheaply. As an added benefit, the cost of the DS1210 is only about five bucks.

The DS1210 has three main functions. Although you may think that the power goes off instantaneously when you turn off your computer, in computer time, it takes quite a while. The DS1210 watches the +5volts from your computer (connected at pin 8) for a drop of more than .25volts. If it sees such a dip, it assumes that the power is going off and prepares the RAM for data retention. First, it checks to see if you are selecting RAM with the chip select line from the computer logic (connected at pin 5). If RAM is being used, the DS1210 waits for it to finish (chip select at pin 5 goes high). It then holds the chip select output to the RAM (pin 6) at a high level, ignoring any more changes at the chip select input. This prevents the RAM from being "glitched" while power is going down. The DS1210 also connects it's battery (connected at pin 2) to the VCC pin of the RAM (connected to pin 1), to supply enough power for the RAM to retain all the data currently stored in the chip. On powerup, the DS1210 disconnects it's battery, allows chip selects from the computer and you're back in business with all of your original data intact. You may even connect a second battery to pin 7 if you are really worried about losing your data. The DS1210 will then switch to this backup battery if the main battery goes dead (seems like overkill in this application - I didn't use this feature).

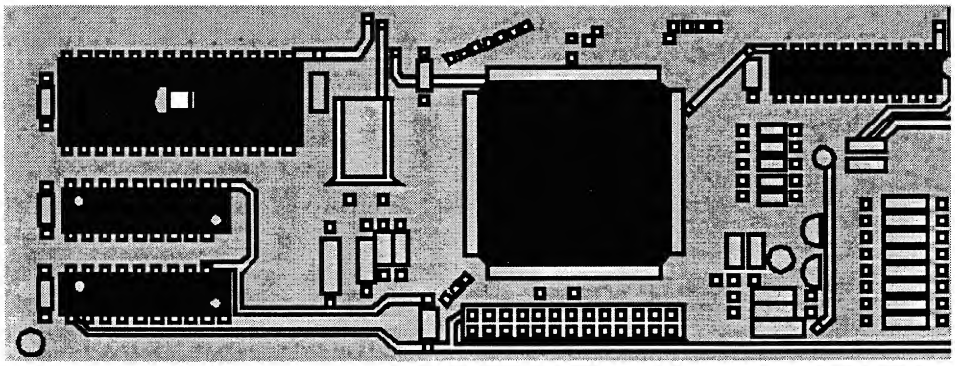
All in all, the DS1210 is just perfect for this application. Many, many power cycles have been executed on my RAM with no problems whatsoever. Although I show a 32Kx8 RAM in the schematic, I actually used a 128Kx8 unit in my controller. With the DS1210 I could use any size RAM I wanted.... may even do a 512Kx8 ramdisk with this guy. The DS1210 worked fine and, it's a little bit cheaper than a Smart Socket. There are also other versions of this controller chip that will control up to 8 RAM chips - for those really big projects. Now, you have two ways to replace your EPROMs and ROMs, so get busy and hack a new 1050 or something!



# Smart IDE

by Bob Woolley

*After building and running the SmartOS in last year's issue, I decided that a few things were not to my liking.*



## Out with the old

Top of the hit list were all those switches. I just don't like to drill a bunch of holes in my 1200XL case.... And, I keep leaving them in the wrong position. And, I can't remember which way is up. And, it is an ordeal to load the RAM - lots of load this and change this switch and then load that and change that switch and push this button. Out went the switches. Also, the programmable internal cartridge just adds a lot of complexity with little benefit - out it goes, too. With these changes, we can now load an OS image in a single, self-executing step. Memory writes can be enabled or disabled from software and two complete SRAM OS versions (as well as the stock ROM OS) are available electronically.

## In With the New

Refer to the Smart IDE schematic. The switches have been replaced by an 8-bit latch (U5) that is selected at memory address $D180 by the HC138 decoder (U4). The bits in this latch are all set to zero on powerup or RESET and control four functions: override, ROM select, write enable, and RAM bank select. The default settings are therefore, no override (bit 7=0), SRAM select (bit 6=0), write disabled (bit 5=0). and SRAM bank 0 (bits 0-2=0). Lets discuss these functions one at a time.

## Bit Stuff

### 6

Bit 6: There are always those programs where any OS changes cause it to crash. Seems like some people have to tweak just about everything they can... When this happens, the only recourse is to use the stock OS and this is the function of bit 6. Although the default setting is for SRAM, you can have the SRAM OS set bit 6 at $D180 and JMP to RESET at $C2AA. This will put you into the ROM OS on powerup and any time you hit the RESET key. To return to normal, just reload another SRAM OS.

### 7

Bit 7 (we took the bits a little out of order to make it easier to understand) Having no switches is very nice for loading a new OS. Problem is - if you get an OS in there that will not run, how do you load a new one? The default, powerup state is from the first SRAM bank. If that is trashed, how do you fix it? Hmmmmmm.... No switches. OK, how about a joystick? Plug a joystick into port 2 and hold down the fire button. This forces you into the ROM OS in the 1200XL, which will always work. Store $40 into $D180 (bit 6, above) and you are latched into the ROM (let go of the button). Works great! When you are using SRAM, just store $80 into $D180 (bit 7) and the joystick fire button has no effect (so you can use the joystick normally). This is the function of the override bit which should normally be set on somewhere in your SRAM OS code.

### 5

Bit 5 This is the write enable function that causes so much trouble in the switch version, since having writing enabled while you are running seems to cause frequent data overlays. When bit 5 is zero (default), you cannot alter any of the SRAM data. Any routine that needs to write to SRAM must first store $20 into $D180, which sets bit 5 on. It is good practice to set it back to zero when you are done! This allows you to load an OS image from disk (or move it from ROM using MOVEROM2.OBJ), enable SRAM writing with bit 5, and move the new image into one of your SRAM banks (using MOVERAM2.OBJ) - all in one operation.

### 0-2

Bits 0-2 These bits select which 16K OS segment you will be using in the $C000 to $FFFF address range. Although the schematic shows a 32Kx8 SRAM, I used a 128Kx8 device in my latest unit. So, instead of 2 OS banks, I now have 8 selections - all controlled by bits 0-2. (If you wire in a 128Kx8 SRAM, the A15 and A16 address bits from the chip go to bits 1 and 2 of the HC273 latch) If you really want to get crazy and use a 512Kx8 SRAM, bits 3 and 4 are available.... (32 banks?)

## Result

So, what did we end up with here? The SmartIDE is designed to allow you to run multiple customized operating systems in your 1200XL and the address range at $D600 - $D7FF has been enabled to provide usable space for new tables and software drivers. These features give you the ability to easily modify the OS in your 1200XL and the extra 512 bytes are a great place to run ramdisk drivers, trace tables, mouse controllers, or keyboard

function keys routines. (you may even see some or all of these in future issues....)

## Building It

The construction of the original SmartOS used the OS ROM sites on the 1200XL to supply power and signal lines. This IDE version uses the CPU socket instead, since the 6502 is more centrally located and allows a much larger board to be used. The SmartIDE uses what I call a rip-off board layout - one of the ICs on the motherboard is removed and it's socket is used to mount the new board full of components (see figure 1). This provides two major benefits: wiring from the motherboard to a daughterboard tends to be messy and fragile and it is easy to move your creation to another 1200XL if you have a reason to do so.

## A Learning Experience

As an example, (now, pay attention - you won't want to learn this the hard way folks....) one of my SmartIDE 1200XLs did not work after I plugged in a 65816 CPU.

OK, so maybe we have a problem with the Sweet16 and my design.

Nope, actually, I had demonstrated this machine at an SLCC meeting and it did not run one of the DOM selections properly. Since the IDE is fairly early in the development cycle, I assumed that it was a design problem in the software. But, when I tried it the next day at home, it worked just fine. Hmmmmmmm. After another problem appeared in an area that used to work, I decided it was not a design problem - the 1200XL was broken! I pulled out another 1200XL motherboard, modified it for the SmartIDE and tried the board in that system. Now, everything works as it should - Sweet16 and all. Very valuable lessons. Make it easy to move a 'project' and thoroughly test anything before you modify it or you'll take a trip down the garden path.

## On Track Again

Back to the daughterboard. I used an old RadioShack project board that I do not find in the catalog. You can use the 276-147 and trim off the excess. The terminal strips are the most difficult part of any rip-off project. These are .5 inch long, .018 dia. double ended pins that connect your board to the socket on the motherboard (see figure 1). They are not easy to find.... Do not use .025 square pins for this purpose - they are too large for the socket and will distort the socket pins. If .025 is all you can find, replace the socket with something that will accept an .025 pin properly. I found a supplier for Samtec part no. TS-132-G-AA pins some time ago and bought what I thought was a lifetime supply. I guess I'm living longer than I thought because I'm running low on pins. The Samtec company is still around (look on InterNet). Last phone number is 812+944-6733. Ask them for a distributor near you. If you really get stuck, I may have better information after I try to get some more for my stock (a number of manufacturers make these kinds of things - look around).

## Easy Parts

The rest of the components are somewhat easier - IC sockets (put all your ICs in sockets), 40 pin DIP header, 10 pin DIP header, ICs and such. Mount them neatly on the daughterboard and wire the circuit as shown in the schematic. Take a look at the upper left corner of the SmartIDE schematic (at D8). The eight circles are the connector to the motherboard that connects us to those signals not available on the 6502. I used a 10 pin IDC connector and a flat cable soldered directly to the motherboard. This still allows me to completely remove the daughterboard by disconnecting the 10 and 40 pin connectors (at D8 and C1). The six signals called out are:

| | |
|---|---|
| **D1** | U18 pin 14 |
| **D6** | U18 pin 9 |
| **D7** | U18 pin 7 |
| **OS** | right side of W12 (remove W12 jumper) |
| **CSROM** | left side of W12 |
| **J/S** | right side of R62 (below cart. port) |

As I said, these wires are soldered directly to a 10 conductor flat cable which plugs into the 10 pin header on the daughterboard.

## More Ripoff

Look closely at figure 1. The rip-off

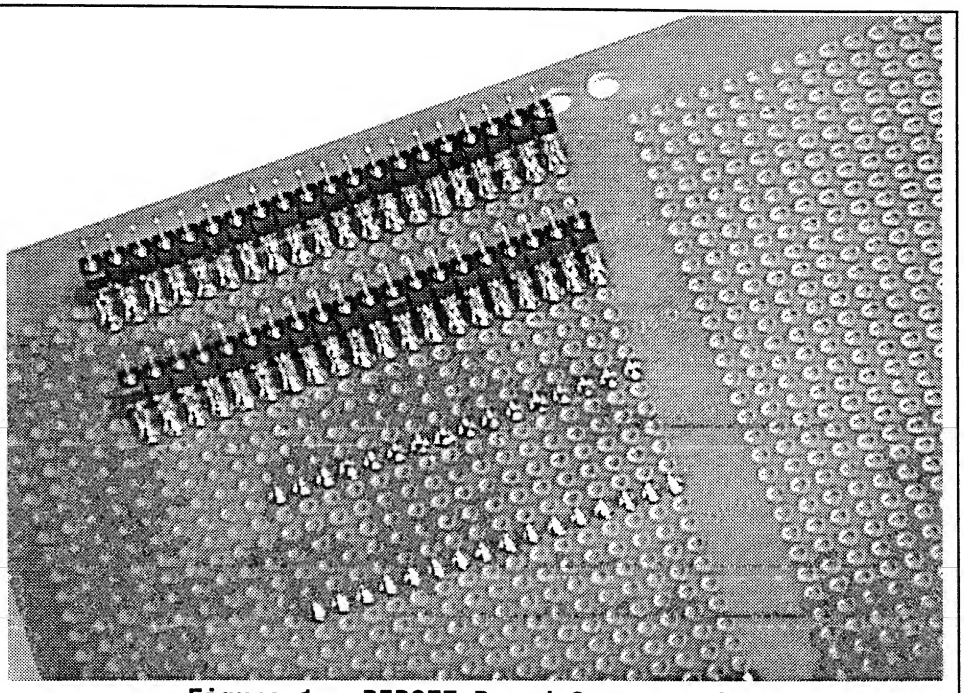


**Figure 1 RIPOFF Board Construction**

technique is to offset the new socket 0.1 inch from the original position (which now holds the terminal strips soldered to the bottom of the daughterboard and plugged into the original 6502 socket on the motherboard). This gives us pairs of pins that must be bridged with solder to reconnect the original signals. Carefully melt solder onto both pins in each pair, using your iron tip as a link between the two pins. With a little practice, you will be able to create a solder short between the two pins quite easily - much faster than using little wires.... If you are not going to connect all the original circuit to the new 6502 location, do not bridge that pair of pins, OK? (this project connects all the pins).

## Soldering Tricks

When you are connecting more than one wire to a solder point, you may notice that the second soldering operation melts all the solder on the pin and the first wire pops out of place. It takes a little practice and a properly set iron temperature, but you can solder to just half of the pin. Set your iron to the lowest temperature that will melt the solder. You need to do this so that the whole pin will not melt at once. Solder the first wire on one side or the other of the pin. The second wire will be soldered on the opposite side of the pin, holding the wire between the pin and iron. This will heat the wire first, which will eventually begin to melt the solder on the pin through the wire. This is not proper soldering procedure, of course, but it does work well when you only want to solder half of a pin. After a second or so, the solder will begin to melt on the pin and the wire will sink into the solder. Once this happens, remove the iron. You will find that only about half of the solder on the pin has melted and the second wire is well connected, while the first wire has not moved. Practice.....

## Getting Off the Ground

Once you have built the hardware, you must make an initial load into the SRAM. The circuit is designed so that bank 0 of the SRAM is enabled and write protected after poweron or RESET. At this point, SRAM has no useful data in it - if you just power on your 1200XL, you will get a blank screen. Get out your joystick and plug it into port 2. Or, make a small jumper and connect pins 6 and 8. Plug in your ED/ASM cartridge, hold the trigger button down and power on the 1200XL. Still holding the trigger, go to BUG and store a $40 into location $D180. Now, you can let go of the trigger. The system will stay in the OS ROM until you zero bit 6 again at $D180.

## ROM Image

If you have a ROM image of an operating system on disk, you can load it into memory at $4000 now. You cannot just load it into $C000 since you are a) in ROM at the moment and b) SRAM is write protected. (you will be able to create a loadable image if you follow the examples later in the article) If you do not have an image on disk, move the stock OS in ROM into $4000 as a starting point. To do this, load MOVEROM2.OBJ into memory from the ED/ASM editor. Go to BUG and G600, which runs the routine. Once finished, you can display $4000 and confirm that it now holds the same data as $C000. Either way, you are now ready to load SRAM.

## Into RAM

You can now load MOVERAM2.OBJ from the editor. Go to BUG and G600 again. This will move the OS from $4000 into the first bank of SRAM. From this point, you can power off or RESET - SRAM is loaded in bank 0. The same procedure may be used to load the other SRAM banks, but you must alter location $60E with the desired bank number before you run the routine.

## SoftWhere

```
Note:
MOVEROM2.ASM is unchanged from the
version in the SmartOS article
(MVROM.ASM) and will not be listed
again.
```

```
0100 ; *** MOVERAM2.ASM  8/02/95
0110 ;     MOVES MEMORY INTO RAM
0120 ;
0130       *=  $0600
0140 ;
0150       PHP
0160       SEI
0170       LDA $D40E
0180       PHA
0190       LDA #$00
0200       STA $D40E
0210 ;
0220 RAMON LDA #$A0
0230       ORA #$00
0240       STA $D180
0250 ;
0260       LDA #$00
0270       STA $CB
0280       STA $CD
0290       LDA #$C0
0300       STA $CC
0310       LDA #$40
0320       STA $CE
0330       LDX #$7D
0340       LDY #$00
0350 ;
0360 LP1 JSR MOVEPAGE
0370       BEQ COMPLETE
0380       CMP #$D0
0390       BNE LP1
0400       LDA #$50
0410       STA $CC
0420 ;
0430 LP2 JSR MOVEPAGE
0440       CMP #$58
0450       BNE LP2
0460 ;
0470       LDA #$D8
0480       STA $CC
0490       BNE LP1
0500 ;
0510 MOVEPAGE LDA #$FD
0520       STA $D301
0530       LDA ($CD),Y
0540       STX $D301
0550       STA ($CB),Y
0560       INY
0570       BNE MOVEPAGE
0580 ;
0590       INC $CE
0600       INC $CC
0610       LDA $CC
0620       RTS
0630 ;
0640 COMPLETE PLA
0650       STA $D40E
0660       PLP
0670       LDA #$FD
0680       STA $D301
0690 ;
0700 RWOFF LDA #$C0
0710       STA $D180
0720       BRK
0730       .END
```

# The IDEal Hard Drive

by Bob Woolley

*So, you got the SmartIDE working? Want to add an IDE hard drive? OK. Let's do it.*

## Inside or Out?

You have two options: mount the drive inside your 1200XL or leave it outside. There is very little to do if you mount it external to the 1200XL and not too much extra to put it under the covers. I prefer it inside myself, so I'm going to lay it out that way for you. At some point in time, I'll do an IDE only upgrade where you don't need all the SmartOS hardware and the IDE code can be burned into the OS ROM.

## Little Enuf

Refer to the SmartIDE schematic. The only additional parts needed to add an IDE drive are the 40 pin IDC header and the two 75 ohm resistors on U4. Piece of cake. Just be sure to mount the resistors as close to U4 as possible and keep your 40 conductor cable under 24 inches. The rest of the modification deals with mounting the drive.

## In the 1200XL

Although I used a 2.5 inch drive in the unit on our cover, there is enough room for the more common 3.5 inch HDs. Either way, the internal Atari power components have to go. This includes the regulators, filter capacitors, rectifier and the heat sink. You must also remove the modulator, and the AC power input connector.

## Pulling Teeth

First, the modulator. This is a bear to remove since it is soldered to very large pads at multiple locations. Best to use one of the big 40 watt irons for this like a RadioShack 64-2071. The way I wrestle the modulator is to slide a thin screwdriver blade under the case while heating one of the outside lugs. When the solder melts, pry the modulator away from the board and allow the solder to solidify again. This usually results in a 1/16th inch gap between the board and the bottom of the modulator. What you are doing is bending the motherboard away from the lug so don't go kingkong here. Now melt the other outside lug and pry it up a little. Cut the four signal leads at the back of the modulator and work the two rear lugs out of the board. Eventually, you will be able to completely remove the front two pins and the job will get easier. Take your time and be patient. Don't crack your motherboard! Clear the four signal pins at the rear. *** very important *** Add a jumper from the last pin (marked with a 1) to the ground surface that was under the modulator. Refer to the photograph of this step. The modulator is used as a ground distribution conductor and if you don't add the wire, you get a terrible video problem!

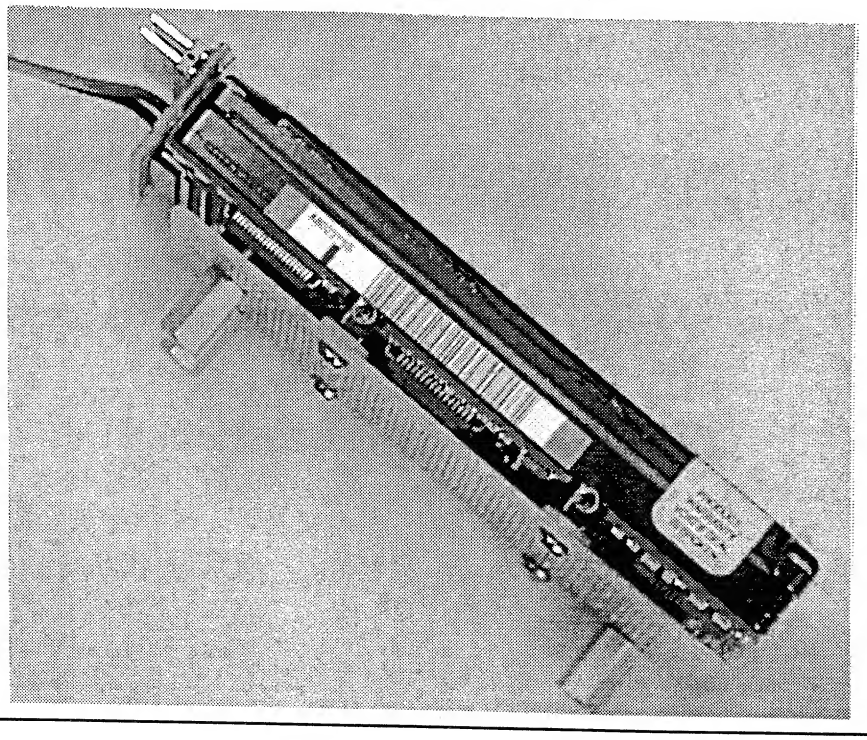
## New +5v Input

Next step is to remove the 9vac power connector - just unsolder it with a 40 watt iron. I then soldered in a two pin polarized connector that will serve as the +5 volt input for the 1200XL. Orient the connector so that the ground is on the left and the +5 volts is on the right.

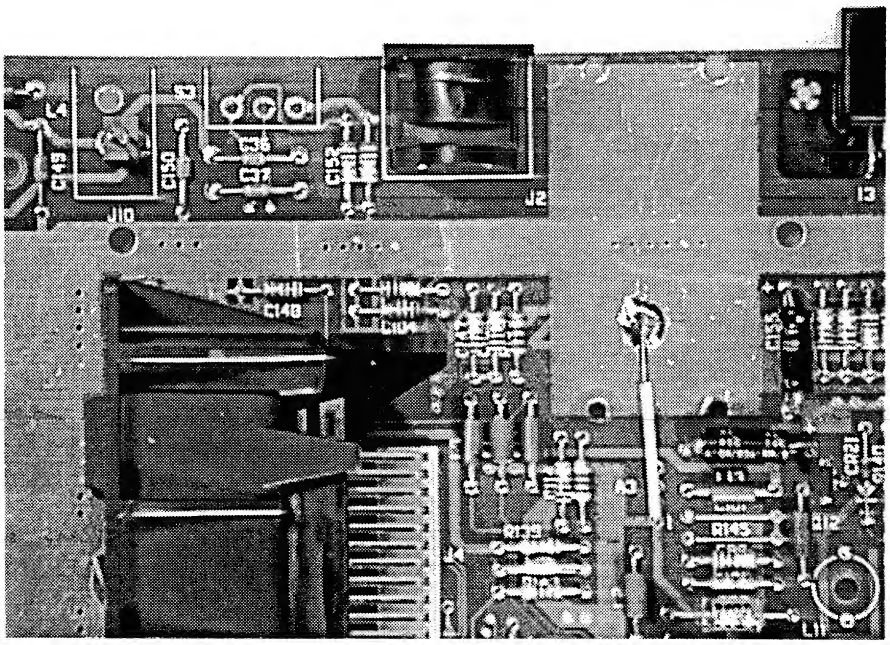
## Old Power Parts

Now, the power. Pull C39 (the large filter capacitor) and cut the legs off of the rectifier (CR12). Disconnect the regulators at A1 and A2 and demount the heatsink entirely. Clear out all the solder from the removed component mounting holes and add 22 gauge or larger jumper wires to the motherboard. The jumper starts at the far left pad of the rectifier CR12 (check the detail photograph). It connects to the right-most pad of the A2 and A1 regulators (check the detail photograph again). What we are doing here is providing a path from the power connector that we added to the +5 volt buss on the motherboard. Now add a wire from anode of CR11 to the anode of CR9 (yep, check the detail photograph). This takes care of ground. You should now

2.5 Inch HD Mounting



Modulator Mount After Modifications



have a solid connection from the left hand pin of your new power connector to the 1200XL ground buss and a solid connection from the right hand power pin to both +5 volt 1200XL busses. Check them a couple of times and be sure!

## New Power Input

In the space left vacant by the modulator, I installed a panel mounted 6 pin DIN plug that will serve as an external power input (be smart and use a power connector that will NOT fit in your monitor output jack...). Just ream out the modulator connector hole and screw in the DIN socket. Make sure you offset the new, larger hole high enough to clear the motherboard. This power plug will supply +5 volts to both the motherboard and the IDE drive as well as +12 volts for the drive. Wire a four pin AMP connector (for the drive) and a compatible +5 volt connector to this socket (see detail photo). Check everything three times before you apply power to this, OK?

## Mount the Drive

Last, but not least, mount the IDE drive on a piece of perfboard and add standoffs for the motherboard. I used a Shack 276-158 board that just happened to be a perfect size for the job. Nothing critical here, mount the drive in the cover if you like, but there are some existing holes over in that corner of the 1200XL (where the shield used to mount) that make it a drop-in.

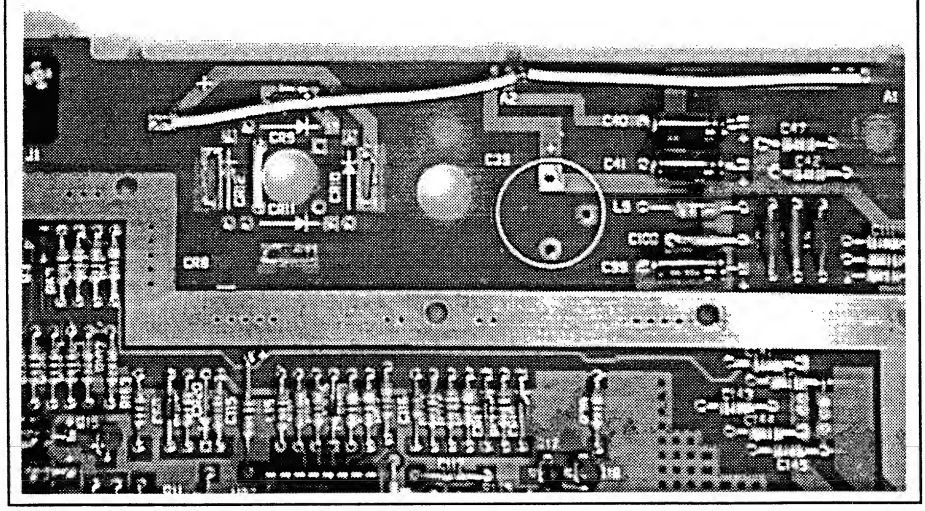
## Do Your Own Thing

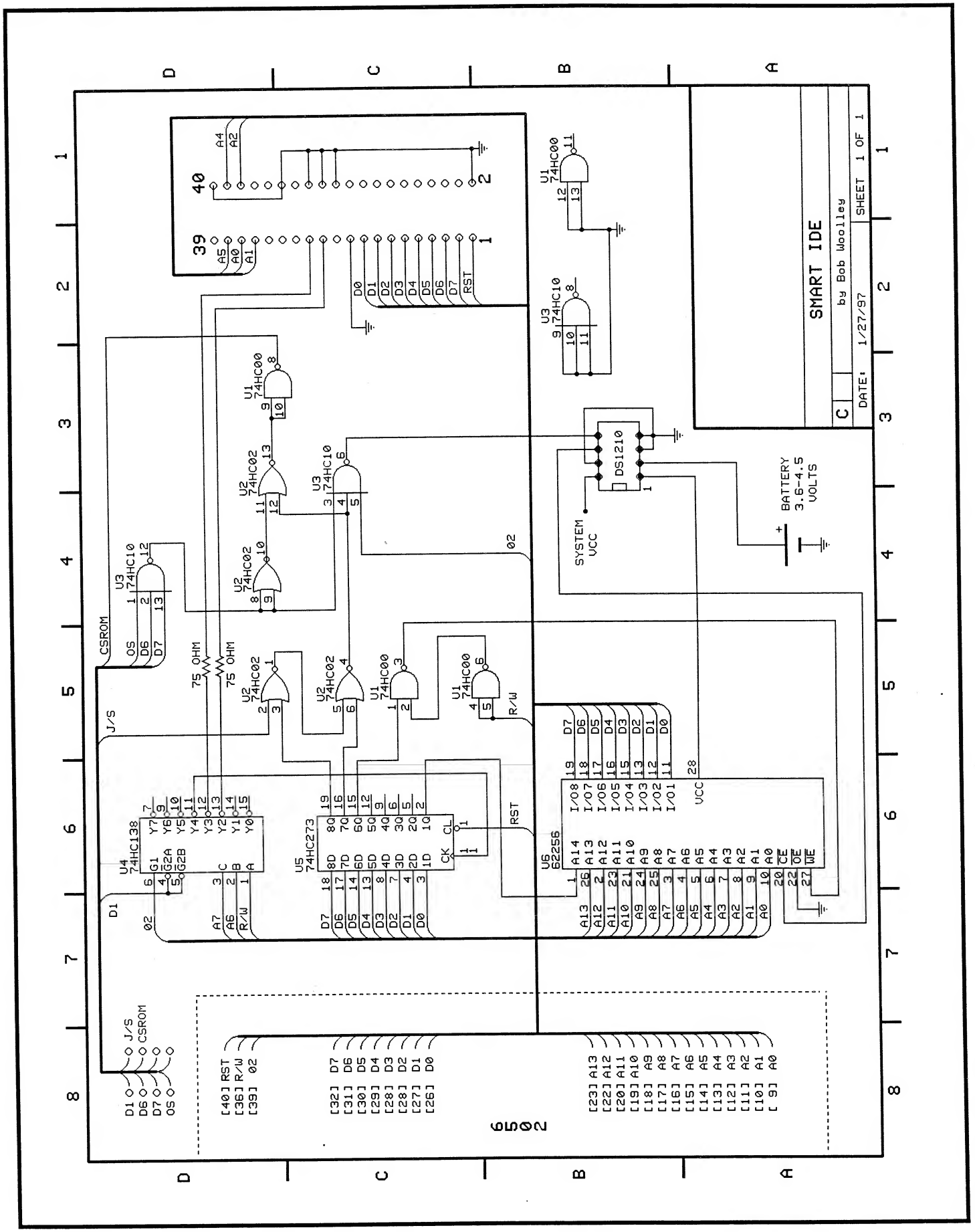
You need not make your physical layout match mine - just provide yourself with an easy to handle, clean configuration. For example, my 1200XL motherboard is still easy to remove since the power just plugs into the board and the DIN power connector is high enough for me to slide underneath it. Since the drive is mounted to the motherboard and not the case, the whole assembly comes out as a unit. I do use an external drive on one of my other IDE 1200XLs since it mounts a SyQuest EZ-135 cartridge drive. I probably could have mounted the EZ-135 inside also, but I like to be able to see the drive when I change cartridges. So, use a configuration that suits you - just be neat.

Oh, the power supply. If you mount the drive externally, the 1200XL just uses it's old Atari supply. If you choose an internal mount as I have shown, use a small switching supply or a good linear unit to power both the IDE drive and the 1200XL. One possibility if you use a 2.5 inch drive is an old 520ST external power supply since the 2.5s do not require +12 volts.

OK - all you need is software!

Power Supply Area After Modifications




SMART IDE
by Bob Woolley
DATE: 1/27/97
SHEET 1 OF 1
CSROM
J/S
R/W
02
RST
SYSTEM VCC
BATTERY 3.6-4.5 VOLTS
75 OHM
75 OHM
DS1210
74HC138
74HC273
62256
74HC00
74HC02
74HC10
6502

# IDE Driver Software

by Bob Woolley

*The last piece of the puzzle - the software. Take a look at the listing. See the title? IDE38.ASM That is the 38th version of the IDE code that I have done. And, it will not be my last. Let's talk about how an IDE drive is programmed.*

## IDE Overview

One of the great things about an IDE drive is that it very simple to run. No low-level formatting is required, you tell it how many heads and sectors per track (not the other way around), and you only need two instructions, read and write! You communicate with the drive using two sets of registers - in our case at $D15x and $D166. You load the sector number(s) into the proper registers and just ask the drive to read or write. If you are using a late-model, buffered drive, data is read or written as fast as you like from the data register until the transfer is complete. From an 8-bit standpoint, the real deal here is that all sectors are 256 bytes. No double read buffer gyrations needed. Of course, there is a loss of half of your drive's capacity since the interface is 16 bits wide and we're only using 8..... Yep, you only get to use 40 megs from an 80 MB drive. Or, 600 megs on a 1.2 GIG unit. SO? That's the price you pay for such a clean, fast, and simple design. At today's prices, who can complain? (don't even ask me about using the whole drive - I am not going to do a 16 bit interface. It would be slow, expensive and messy.)

## Reset

OK, on to the registers. First is $D166. This is the reset register. The interface has hardware reset from the 1200XL connected to pin 1 of the drive. Every time you push RESET on your 1200XL, the drive gets reset (you may even notice the drive start to spin down). When you are doing IDE development, you may want to disconnect the hardware reset and just use the soft reset at $D166. This will allow you to reset only your 1200XL if the system is hung, preserving the IDE register status for examination. Either way, you must assert reset for at least 5 microseconds to clear the drive. It will then take a little while for the drive to get itself back together and go not busy (all regs will read $D0 during reset). Currently, I am not using $D166 reset, so don't look for it in the IDE38 code.

## Registers

Next are the $D15x regs. I'll list them in order:

| ADDR | READ | WRITE |
|---|---|---|
| 50 | data reg | data reg |
| 51 | error reg | write comp |
| 52 | sector count | sector count |
| 53 | sector number | sector num. |
| 54 | cylinder low | cylinder low |
| 55 | cylinder high | cylinder hi |
| 56 | SDH | SDH |
| 57 | status | command |

Most of those may be self-explanatory. Data is the read or write data. The error reg has error codes if $57 has bit 0 set (xxxxxxx1). Sector count allows you to read multiple sectors with one command - set it to 1. Sector number is the number of the sector you wish to read or write ($01 - $10). Cylinder low and high are the cylinder number you are accessing ($0000 - $FFFF). SDH contains three parameters: the sector size in bits 5 and 6 (01), the head number in bits 0 - 3 ($0 - $F), and the drive (master or slave) in bit 4. For example $A7 selects the master drive and head 07. $B9 selects the slave drive on head 09. (I have been able to run two drives at once with some success.... but it needs some s/w yet).

## Code Overview

The driver code and the configuration table sit in the space newly acquired at $D600 thru $D7FF. This may allow some software that poaches in the OS space for its own routines to run with an IDE drive, since $D6xx and $D7xx are never usable from a software routine. The IDE drivers will run with a DOS that is not HD intelligent. Many of my partitions boot good old DOS 2.0, both single and double density. Just the ticket for AtariWriter on cartridge. In fact, the IDE drive runs without any DOS at all, as in SCOPY and other m/l utilities. It

just looks like any other disk drive to the SIO code. There is currently space for 16 partitions in HDTABL, but 63 would be supported in the current code. And, the HDTABL could be loaded from disk as many times as necessary, giving you unlimited partitions - 16 at a time. No or little error checking is done in IDE38 code. This is NOT a great idea. While I am using new, modern drives, I don't seem to have a problem. Some of my old, high-mileage clunkers do not fare so well. The older drives do not seem to like having 16 heads and 16 SPT, either. So, beware. This IDE stuff is not all perfect in the software department. Of course, the value of the SmartIDE modification is to develop OS routines easily - in particular, the IDE code. So, you folks get on the ball and do some of this stuff. Write some good hacks to the code and send it to AC for future articles! Once we have some really effective code you can add an IDE with just the HC138 and a new ROM. The hardware seems to be pretty solid. That is not to say that it does not have any problems - the Sweet16 will not work with it, I'm afraid. But, on a standard machine, it is at least consistent and reliable.

Ok, on to the code itself.

## IDE vs. SIO

The IDE drive has a hook into the normal SIO code at $C95B. We JSR into the new routine to see if the I/O is pointed to an IDE partition. The first check is the SELECT key (line 360-380). I have allowed the SELECT key to override the IDE code. If you want to bypass all of the active IDE partitions, just hold SELECT down and your system will use only SIO devices. This makes it quick to peek at a directory or copy a single file to or from a floppy. For example, you have a floppy set as D2: and an IDE partition set as D2:. The IDE drive at D2: will always answer for D2: since the IDE code has priority. But, if you want to read from the floppy D2:, you can simply hold SELECT down and all D2: access will be on the SIO. To copy a file from the floppy D2: to the IDE D2: you can hold SELECT when DOS asks for the source and release SELECT when DOS asks for the destination drive. Works just fine. The next test is for the device in the OS DDEVIC ($0300). If it is not a $31, then we are not talking to a disk drive and we can exit IDE code. The last test is to find the device address, DUNIT ($0301) in the HDTABL at $D7C0.

## HDTABL

The HDTABL is a 64 byte data field that contains 16 four byte entries. Each entry represents a partition or range of sectors on the IDE drive. The first byte is the high order byte of the first sector in the partition and the second byte is the low order byte of the first sector in the partition. The third byte is the number of 256 sector clusters in the partition (max sectors is still 65K or 16mb). The fourth byte is a composite - bits 5-7 are density tags that are returned on a STATUS command. $8x is a 1050 in dual density (1024 sectors). $6x is a double density disk. And $0x is a single density disk. Bit 4 is the IDE drive bit (which I do not use in the current code). Setting bit 4 will select the slave drive on the IDE buss instead of the master drive. This will be a great help when you want to back up your HD data on one of those SyQuest 135mb removable drives. Just add it as a slave drive and copy from one IDE drive to another! Finally, bits 0-3 are the drive address itself. As an example, a $64 in byte 4 would indicate a DD partition on the master drive that will respond to D4: on the SIO. A $92 would be a 1050 density partition on the slave drive that would answer as $D2:. Any drive that has bits 0-3 set off is not active or disabled. Examine the data loaded in the HDTABL in IDE38.ASM. The first partition at line 2730 is a DD drive set as D1: on the master drive. It is 4096 sectors long and starts on sector $0000 of the IDE drive. The entry at line 2760 is a SD drive that is unassigned and is 1536 sectors long starting at IDE sector $001C. And, as I write this I notice that the next partition starts at IDE sector $0020 - which does not allow enough space for the previous partition! You be very careful when you set up HDTABL, OK? That little error in line 2770 would eat my 8 meg partition if I used the previous partition past 1024 sectors, wouldn't it? Note that the HDTABL is not required to be in any specific order except for the first three entries, which we will cover later. This allows you to overlay any of the entries you wish in order to change which partitions are "mounted" on a particular "drive".

## Searching HDTABL

Finding the device address in HDTABL is just a matter of checking byte four of each entry for a match to $0301, DUNIT. Line 420 loads the an index to the end of the table and the search progresses backwards until a match is found or the index is less than zero. If no match has been found, then that drive is not an active IDE drive. The first drive found starting from the bottom will cause a branch into IDE code at line 580.

## R We Busy?

Before we try to access the drive, we check for BUSY at line 590. This code just loops forever if the drive is hung. A better approach would be to branch out into a routine that would determine the reason for the BUSY and take action. Maybe later.... As I said earlier, a new drive never has a problem here. Older drives, not so true. OK?

## Command Decode

Now we are into the real code at line 630. Here we determine the command. No real cute coding; just compare DCOMND at $0302 with all valid IDE commands. If we do not find one, return a command reject at line 790. Notice that the $4E command in line 760 just branches into the STATUS routine. This command and the $4F command are the read and write configuration commands used by the PERCOM standards. If you want to use SpartaDOS or any other DOS that needs this data, you must implement these two commands. And, yes that means that any DOS using these commands will not recognize the IDE drives as anything other than an 810 or 1050. Gives us something to work on, doesn't it? MYDOS works just fine without the $4E and $4F.

## SETREGS

So, now we get to the command itself. The only commands that do anything with the IDE drives are the read and write commands. The first step in both commands is to JSR to SETREGS. This is where we convert the logical sector number from SIO to a physical sector, head and cylinder on the IDE drive. Lines 1540 to 1580 sets the data address for the transfer. Lines 1590 to 1630 loads the IDE sector number ($D153) from the low order SIO sector number ($030A). Notice how simple that is? Later on, we'll discuss the IDE INIT code where we tell the IDE drive that he has 16 sectors per track and 16 heads per cylinder. This makes it simple to convert the SIO sector number to the IDE parameters. Lines 1640 to 1710 loads the IDE physical head ($D156) from the remaining low order SIO sector value. Lines 1720 to 1790 adds the HDTABL partition offset in bytes 1 and 2 to the high order SIO sector ($030B) and stores the result in the IDE cylinder low ($D154) and cylinder high ($D155). It also checks the range of the SIO sectors at line 1730 to insure that we don't overrun the next HDTABL partition. Last, but not least, we tell the IDE drive that we will only read one sector (lines 1800 to 1810) and clear the carry bit to signify no sector error.

Back in the read or write command, we check for a sector overrun and start the command.

## Read Command

The IDE read command is a $20. Just loading that value into the command register at $D157 will initiate the seek to the data sector and read all the data into the IDE buffer. Once the drive is not BUSY (lines 1190 to 1210), you are ready to read your data. By checking the data ready bit (bit 3) in the STATUS register ($D157), we can determine when the sector may be transferred into our memory (at ($32),Y). Once the first data byte is ready, by the way, it is not really necessary to check status to see if any subsequent bytes are available since we are reading a memory buffer, not a mechanical device at this point. There are two sector sizes on the Atari, 128 and 256 bytes, as defined in DBYTLO at $0308. If we are reading a 128 byte sector, the last 128 bytes just get thrown away (lines 1300 to 1340). We do not want to read them into memory. A 256 byte sector will read all 256 bytes into memory (lines 1220 to 1290). At that point, bit 3 of the IDE STATUS register will be reset to 0. That's it. Set a few flags in the SIO and you're done (lines 970 to 990 and 840 to 880). Cut out all the unnecessary code and you can read a 720 sector DD disk in about 4 seconds..... not too bad!

## Write Command

The write command is similar to the read command except the command code is a $30. We do not need to check the sector size here since the last half of the sector is not used for 128 byte sectors anyway. The code simply writes the 256 bytes starting at the SIO data address into the IDE data register. The drive will do the seek and store the data after the transfer. Be aware! As written, this code does not keep you from writing 256 DD sectors to a SD drive as would normally be the case. You can hose yourself all up if you are not careful in the HDTABL configuration. It would be a good idea to check the sector length against the HDTABL size..... Maybe later.

OK, other stuff....

## IDE Initialization

IDE drives must be initialized before you use them. You must tell the drive how many sectors per track and how many heads per cylinder they should configure themselves into or the drive will use it's native values. We want to have a drive that has 16 sectors per track and 16 heads - this makes the code very simple. So, we do this every time we RESET the drive. Line 320 patches the OS to init the IDE drive. INITCODE (at line 1850) first checks for the IDE drive to have completed it's RESET and then stores the number of sector in $D152 and the number of heads in $D156. Storing a $91 in the command reg at $D157 initializes the drive with our values (line 1940 to 1950). Simple.

## Dynamic Selection

There are a couple of bells and whistles in this code that I like. The first three HDTABL entries are reserved for D1:. I have patched the OS to respond to CNTL-F1 in lines 180 to 200 so that this keystroke now selects D1: from one of four options. SIO D1:, if it is connected, or any of the first three HDTABL entries. The purpose of this feature is to allow you to change "diskettes" in D1: on the fly. Or, you can switch from the IDE drive to the SIO drive with just a keystroke. I also use the two L1 and L2 indicators on the 1200XL to indicate just who's on D1:. 00 is the SIO drive, 01 is HDTABL entry 1 (at $D7C0), 10 is HDTABL entry 2 ($D7C4), and 11 is HDTABL entry 3 ($D7D0). This code is in SELNXTHD at lines 205 thru 2100 and SETL1L2 at lines 2430 thru 2540. I also allow D2: to be toggled between the HDTABL IDE drive and the SIO drive using the CNTL-F2 keystroke (lines 220 thru 260 and 2590 thru 2680). Overall, I can select IDE drives 1-3 or the SIO drive on D1: and IDE drive or SIO drive on D2: from within any program that has not overlaid the OS too badly. Works really great!

## FINIS

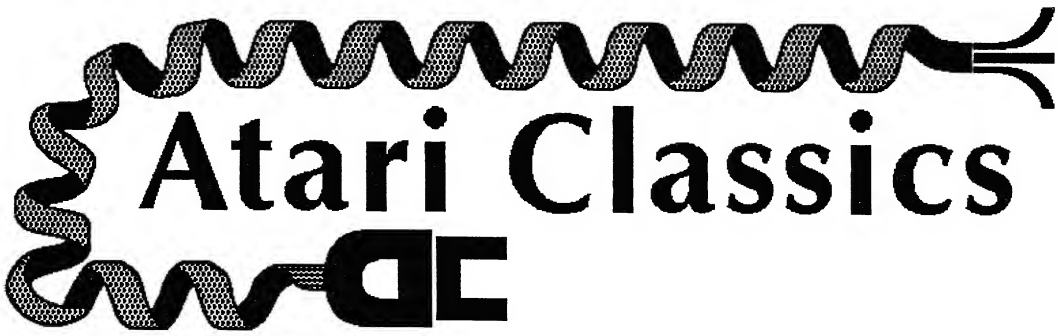
Anything else? I guess not in this issue. I will really, really try not to screw up the publication schedule any more. I know a number of you subscribers are wondering about our viability - I would rather print something useful on an irregular basis than just print whatever we have on a fixed schedule. I hope at least some of you agree and continue to support us. For the others, we'll work on it, OK? Don't give up - better yet, volunteer! We could use the help. That way, I may not have to have either a project or a magazine - we can have both.

```
0100 ;IDE38.ASM   06/03/96
                 0110 ;
0000             0120          *=   $C95B      POINT SIO TO IDE
C95B 2000D6      0130          JSR  $D600
                 0140 ;
C95E             0150          *=   $FC98      COLDSTART
FC98 4C77E4      0160          JMP  $E477
                 0170 ;
FC9B             0180          *=   $FC34      SELECT IDE SEG ^F1
FC34 2039D7      0190          JSR  SELNXTHD
FC37 4C6DFC      0200          JMP  $FC6D
                 0210 ;
FC3A             0220          *=   $FCB5      TOGGLE D2 ^F2
FCB5 2091D7      0230          JSR  PROCD2
FCB8 EA          0240          .BYTE $EA,$EA,$EA,$EA
FCB9 EA
FCBA EA
FCBB EA
FCBC EA          0250          .BYTE $EA,$EA,$EA,$EA
FCBD EA
FCBE EA
FCBF EA
FCC0 A900        0260          LDA  #$00
                 0270 ;
FCC2             0280          *=   $C31D      KILL ROM CHKSUM
C31D EA          0290          NOP
C31E EA          0300          NOP
                 0310 ;
C31F             0320          *=   $C400      INIT IDE
C400 2012D7      0330          JSR  INITCODE
                 0340 ;
C403             0350          *=   $D600      IDE CODE
D600 AD1FD0      0360          LDA  $D01F      CHK FOR SELECT KEY
D603 2902        0370          AND  #$02
D605 F019        0380          BEQ  JMPTOSIO   IF DOWN, DO SIO
D607 AD0003      0390          LDA  $0300
D60A C931        0400          CMP  #$31
D60C D012        0410          BNE  JMPTOSIO
D60E A23C        0420          LDX  #$3C
                 0430 ;
                 0440 SRCH
D610 BDC3D7      0450          LDA  HDTABL+3,X
D613 290F        0460          AND  #$0F
D615 CD0103      0470          CMP  $0301
D618 F009        0480          BEQ  IDECODE
D61A CA          0490          DEX
D61B CA          0500          DEX
D61C CA          0510          DEX
D61D CA          0520          DEX
D61E 10F0        0530          BPL  SRCH
                 0540 ;
                 0550 JMPTOSIO
D620 4C71E9      0560          JMP  $E971
                 0570 ;
                 0580 IDECODE
D623 AD57D1      0590 LPBZY    LDA  $D157
D626 2980        0600          AND  #$80
D628 D0F9        0610          BNE  LPBZY
                 0620 ;
D62A AD0203      0630          LDA  $0302
D62D C953        0640          CMP  #$53
D62F F025        0650          BEQ  STATUSCMD
D631 C921        0660          CMP  #$21
D633 F033        0670          BEQ  FORMATCMD
D635 C922        0680          CMP  #$22
D637 F02F        0690          BEQ  FORMATCMD
D639 C952        0700          CMP  #$52
D63B F040        0710          BEQ  READCMD
D63D C957        0720          CMP  #$57
D63F F06D        0730          BEQ  WRTCMD
D641 C950        0740          CMP  #$50
D643 F069        0750          BEQ  WRTCMD
D645 C94E        0760          CMP  #$4E
D647 F00D        0770          BEQ  STATUSCMD
                 0780 ;
                 0790 CMDREJ
D649 A901        0800          LDA  #$01
D64B 8DEA02      0810          STA  $02EA
D64E A98B        0820          LDA  #$8B
                 0830 ;
                 0840 RETURN
D650 8D0303      0850          STA  $0303
D653 A8          0860          TAY
D654 38          0870          SEC
D655 60          0880          RTS
```

```
                 0890 ;
                 0900 STATUSCMD
D656 BDC3D7      0910          LDA  HDTABL+3,X
D659 29F0        0920          AND  #$F0
D65B 8DEA02      0930          STA  $02EA
D65E 8D0803      0940          STA  $0308
D661 8D0903      0950          STA  $0309
                 0960 ;
                 0970 CLRSTATUS
D664 A901        0980          LDA  #$01
D666 D0E8        0990          BNE  RETURN
                 1000 ;
                 1010 FORMATCMD
D668 AD0403      1020          LDA  $0304
D66B 8532        1030          STA  $0032
D66D AD0503      1040          LDA  $0305
D670 8533        1050          STA  $0033
D672 A9FF        1060          LDA  #$FF
D674 A000        1070          LDY  #$00
D676 9132        1080          STA  ($32),Y
D678 C8          1090          INY
D679 9132        1100          STA  ($32),Y
D67B D0D9        1110          BNE  STATUSCMD
                 1120 ;
                 1130 READCMD
D67D 20D2D6      1140          JSR  SETREGS
D680 B0C7        1150          BCS  CMDREJ
D682 A920        1160          LDA  #$20
D684 8D57D1      1170          STA  $D157
D687 A000        1180          LDY  #$00
D689 AD57D1      1190 LP1      LDA  $D157
D68C 2980        1200          AND  #$80
D68E D0F9        1210          BNE  LP1
D690 AD57D1      1220 RDLP     LDA  $D157
D693 2908        1230          AND  #$08
D695 F0F9        1240          BEQ  RDLP
D697 AD50D1      1250          LDA  $D150
D69A 9132        1260          STA  ($32),Y
D69C C8          1270          INY
D69D CC0803      1280          CPY  $308
D6A0 D0EE        1290          BNE  RDLP
D6A2 AD50D1      1300 ENDRD    LDA  $D150
D6A5 AD57D1      1310          LDA  $D157
D6A8 2908        1320          AND  #$08
D6AA D0F6        1330          BNE  ENDRD
D6AC F0B6        1340          BEQ  CLRSTATUS
                 1350 ;
                 1360 WRTCMD
D6AE 20D2D6      1370          JSR  SETREGS
D6B1 B096        1380          BCS  CMDREJ
D6B3 A930        1390          LDA  #$30
D6B5 8D57D1      1400          STA  $D157
D6B8 A000        1410          LDY  #$00
D6BA AD57D1      1420 LP2      LDA  $D157
D6BD 2980        1430          AND  #$80
D6BF D0F9        1440          BNE  LP2
D6C1 AD57D1      1450 WRTLP    LDA  $D157
D6C4 2908        1460          AND  #$08
D6C6 F0F9        1470          BEQ  WRTLP
D6C8 B132        1480          LDA  ($32),Y
D6CA 8D50D1      1490          STA  $D150
D6CD C8          1500          INY
D6CE D0F1        1510          BNE  WRTLP
D6D0 F092        1520          BEQ  CLRSTATUS
                 1530 ;
                 1540 SETREGS
D6D2 AD0403      1550          LDA  $0304
D6D5 8532        1560          STA  $0032
D6D7 AD0503      1570          LDA  $0305
D6DA 8533        1580          STA  $0033
D6DC AD0A03      1590          LDA  $030A
D6DF 290F        1600          AND  #$0F
D6E1 18          1610          CLC
D6E2 6901        1620          ADC  #$01
D6E4 8D53D1      1630          STA  $D153
D6E7 AD0A03      1640          LDA  $030A
D6EA 29F0        1650          AND  #$F0
D6EC 4A          1660          LSR  A
D6ED 4A          1670          LSR  A
D6EE 4A          1680          LSR  A
D6EF 4A          1690          LSR  A
D6F0 09A0        1700          ORA  #$A0
D6F2 8D56D1      1710          STA  $D156
D6F5 AD0B03      1720          LDA  $030B
D6F8 DDC2D7      1730          CMP  HDTABL+2,X
```

```
D6FB B014    1740          BCS  SECTERR
D6FD 7DC1D7  1750          ADC  HDTABL+1,X
D700 8D54D1  1760          STA  $D154
D703 BDC0D7  1770          LDA  HDTABL,X
D706 6900    1780          ADC  #$00
D708 8D55D1  1790          STA  $D155
D70B A901    1800          LDA  #$01
D70D 8D52D1  1810          STA  $D152
D710 18      1820          CLC
D711 60      1830 SECTERR  RTS
             1840 ;
             1850 INITCODE
D712 A950    1860 WAITHD   LDA  #$50
D714 CD57D1  1870          CMP  $D157
D717 D0F9    1880          BNE  WAITHD
             1890 ;
D719 A910    1900          LDA  #$10
D71B 8D52D1  1910          STA  $D152
D71E A9AF    1920          LDA  #$AF
D720 8D56D1  1930          STA  $D156
D723 A991    1940          LDA  #$91
D725 8D57D1  1950          STA  $D157
D728 A240    1960          LDX  #$40
D72A 88      1970 WLP      DEY
D72B D0FD    1980          BNE  WLP
D72D CA      1990          DEX
D72E D0FA    2000          BNE  WLP
D730 2044D7  2010          JSR  FINDX
D733 2076D7  2020          JSR  SETL1L2
D736 4C6EC6  2030          JMP  $C66E      ;OS @ $C400
             2040 ;
             2050 SELNXTHD
D739 2044D7  2060          JSR  FINDX
D73C 2054D7  2070          JSR  INCRX
D73F 2076D7  2080          JSR  SETL1L2
D742 A8      2090          TAY
D743 60      2100          RTS
             2110 ;
             2120 FINDX
D744 A20C    2130          LDX  #$0C
D746 BDBFD7  2140 NXTPARM  LDA  $D7BC+3,X
D749 290F    2150          AND  #$0F
D74B D006    2160          BNE  FOUNDX
D74D CA      2170          DEX
D74E CA      2180          DEX
D74F CA      2190          DEX
D750 CA      2200          DEX
D751 D0F3    2210          BNE  NXTPARM
D753 60      2220 FOUNDX   RTS
             2230 ;
             2240 INCRX
D754 A9A0    2250          LDA  #$A0
D756 8D80D1  2260          STA  $D180
D759 BDBFD7  2270          LDA  $D7BC+3,X
D75C 29F0    2280          AND  #$F0
D75E 9DBFD7  2290          STA  $D7BC+3,X
D761 8A      2300          TXA
D762 18      2310          CLC
D763 6904    2320          ADC  #$04
D765 290C    2330          AND  #$0C
D767 AA      2340          TAX
D768 BDBFD7  2350          LDA  $D7BC+3,X
D76B 0901    2360          ORA  #$01
D76D 9DBFD7  2370          STA  $D7BC+3,X
D770 A980    2380          LDA  #$80
D772 8D80D1  2390          STA  $D180
D775 60      2400          RTS
             2410 ;
             2420 SETL1L2
D776 AD01D3  2430          LDA  $D301
D779 29F3    2440          AND  #$F3
D77B 8D01D3  2450          STA  $D301
D77E 8A      2460          TXA
D77F 4A      2470          LSR  A
D780 4A      2480          LSR  A
D781 AA      2490          TAX
D782 BD8DD7  2500          LDA  DECODE,X
D785 0D01D3  2510          ORA  $D301
D788 8D01D3  2520          STA  $D301
D78B A8      2530          TAY
D78C 60      2540          RTS
             2550 ;
             2560 DECODE
D78D 0C      2570          .BYTE $0C,$04,$08,$00
D78E 04
D78F 08
D790 00
             2580 ;
             2590 PROCD2
D791 A9A0    2600          LDA  #$A0
D793 8D80D1  2610          STA  $D180
D796 ADD3D7  2620          LDA  $D7D3
D799 4902    2630          EOR  #$02
D79B 8DD3D7  2640          STA  $D7D3
D79E A980    2650          LDA  #$80
D7A0 8D80D1  2660          STA  $D180
D7A3 AC01D3  2670          LDY  $D301
D7A6 60      2680          RTS
             2690 ;
D7A7         2700          *=   $D7BC
D7BC 00      2710          .BYTE $00,$00,$00,$00
D7BD 00
D7BE 00
D7BF 00
             2720 HDTABL
D7C0 00      2730          .BYTE $00,$00,$10,$61 ;4096DD
D7C1 00
D7C2 10
D7C3 61
D7C4 00      2740          .BYTE $00,$10,$06,$60 ;1536DD
D7C5 10
D7C6 06
D7C7 60
D7C8 00      2750          .BYTE $00,$16,$06,$00 ;1536SD
D7C9 16
D7CA 06
D7CB 00
D7CC 00      2760          .BYTE $00,$1C,$06,$00 ;1536SD
D7CD 1C
D7CE 06
D7CF 00
D7D0 00      2770          .BYTE $00,$20,$80,$62 ;32K DD
D7D1 20
D7D2 80
D7D3 62
D7D4 00      2780          .BYTE $00,$A0,$10,$63 ;4096DD
D7D5 A0
D7D6 10
D7D7 63
D7D8 00      2790          .BYTE $00,$B0,$06,$64 ;1536DD
D7D9 B0
D7DA 06
D7DB 64
D7DC 00      2800          .BYTE $00,$B6,$06,$65 ;1536DD
D7DD B6
D7DE 06
D7DF 65
D7E0 00      2810          .BYTE $00,$BC,$10,$66 ;4096DD
D7E1 BC
D7E2 10
D7E3 66
D7E4 00      2820          .BYTE $00,$CC,$04,$67 ;1024DD
D7E5 CC
D7E6 04
D7E7 67
D7E8 00      2830          .BYTE $00,$D0,$FF,$68 ;64K DD
D7E9 D0
D7EA FF
D7EB 68
D7EC 01      2840          .BYTE $01,$D0,$30,$60 ;12K DD
D7ED D0
D7EE 30
D7EF 60
D7F0 02      2850          .BYTE $02,$00,$FF,$60 ;64K DD
D7F1 00
D7F2 FF
D7F3 60
D7F4 03      2860          .BYTE $03,$00,$FF,$60 ;64K DD
D7F5 00
D7F6 FF
D7F7 60
D7F8 04      2870          .BYTE $04,$00,$FF,$00 ;64K SD
D7F9 00
D7FA FF
D7FB 00
D7FC 05      2880          .BYTE $05,$00,$FF,$00 ;64K SD
D7FD 00
D7FE FF
D7FF 00
             2890 END
```

## Stay Plugged In To The Atari 8-Bit Scene With:




For many of our subscribers, it is that time again....

Those of you that have a one year subscription (to be defined as a six issue subscription from here on out) are holding their *last* issue! AC has the resources to continue and improve with each issue, but we need your support. Our commitment is to provide you with news, projects and entertainment from the 8-bit world - your commitment is to turn to page 30 and subscribe for six more issues!

### Projected Hardware Articles in 1997 -1998

**IBM Mouse on the 8-bit**
**16 MB Memory Upgrade**
**80 Column Upgrade**
**Secrets of BasicXL**
**IBM Keyboard on the 8-Bit**
**65816 Upgrade**

## Don't Miss Out - Subscribe Today!!

# Atari User Groups

Below are user groups that responded to a North West Phoenix Atari Connection survey, 8-Bit groups that exchange newsletters with the San Leandro Computer Club, all the groups listed in *Feedback*, the newsletter of the Adelaide Atari Computer Club, and those that have contacted us. We believe the North American groups having "?" support both 8-Bit and ST users. We don't know which computers are supported by the Australian groups. You may send additions, deletions or corrections to "Where is..." at AC.

## North American User Groups

**Alamo Area Atari User Association**
AAAUA
Todd Sunrunner, President
P.O. Box 79-1426
San Antonio, TX, 78279-1426
8-Bit

**Atari Bay Area Computer Users Society**
ABACUS
Roger Sinasohn, President
P.O. Box 22212
San Francisco, CA, 94122
8-Bit, ST, IBM

**Atari Exchange of Louisville**
AEL
Jan Wilt, President
P.O. Box 34183
Louisville, KY, 40232
8-Bit, ST

**ATESIG**
ATESIG
Decker McAllister, Jr., President
145 Surf Place
Seal Beach, CA 90740
d.mcalliste2@ genie.com
8-Bit in Model Railroading

**Diablo Valley Atari Computer Enthusiasts**
DACE
Don Birkhimer, President
2834 Rockridge Drive
Pleasant Hill, CA, 94523
8-Bit, ST, IBM

**Front Range Atari Users' Group**
FRAUG
Joseph Michaud, President
3012 Rockborough Court
Fort Collins, CO, 80525
8-Bit, ST

**Garden City Atari Computer Enthusiasts**
1003 Amphion Street
Victoria, B.C., V85 4G2
8-Bit, ST

**Houston Atari Computer Enthusiasts**
HACE
Bill Anderson, President
P.O. Box 820335
Houston, TX, 77282-0335
8-Bit, ST

**Huntsville Atari Users Group**
HAUG
3911 West Crestview
Huntsville, AL, 35816
?

**Indiana-Michigan Atari Group Exchange**
IMAGE
P.O. Box 1742
South Bend, IN, 46634-1742
8-Bit, ST

**Jersey Atari Computer Group**
JACG
Box 406
Berkley Heights, NJ 07922
8-Bit and higher

**Jersey Atari Computer Society (?)**
JACS
818 Drexel Street
Delran, NJ, 08075
?

**L.C.A.C.E.**
L.C.A.C.E.
P.O. Box 8788
Waukegan, IL, 60079-8788
8-Bit, ST

**Miami Valley Atari Computer Enthusiasts**
M.V.A.C.E.
P.O. Box 24221
Huber Heights, OH, 45424
8-Bit, ST, IBM, Mac

**Noah 8**
3632 W. 130th Street
Cleveland, OH, 44111
8-Bit, ST

**North West Phoenix Atari Connection**
N.W.P.A.C.
Dale Wooster, President
P.O. Box 67511
Phoenix, AZ, 85082
8-Bit

**Ol' Hackers Atari User Group**
Ol'HAUG
3376 Ocean Harbor Drive
Oceanside, NY, 11572
8-Bit

**Pinellas Atari Computer Enthusiasts**
PACE
Jean Brokaw, Editor
958 Phyllis Avenue
Largo, FL, 34641
8-Bit

**S.P.A.C.E.**
P.O. Box 120016
New Brighton, MN, 55112
8-Bit

**San Diego Atari Computer Enthusiasts**
S.D.A.C.E.
Paul Blagay, President
P.O. Box 900076
San Diego, CA, 92190
8-Bit, ST

**San Leandro Computer Club**
SLCC
Robbie Bridges, President
P.O. Box 1506
San Leandro, CA, 94577-0374
8-Bit, ST, IBM

**Seattle Puget Sound Atari Computer Enthusiasts**
S*P*A*C*E
John Strand, President
P.O. Box 11042
Tacoma, WA, 98411-0042
8-Bit, ST

**Toronto Atari Federation**
TAF
5334 Yonge Street, Suite 1527
Willowdale, ONT, M2N 6M2
?

## Australian User Groups

**Atari Computer Enthusiasts New South Wales**
A.C.E. (N.S.W.)
Swavek Jabrzemski, President
G.P.O. Box 4514
Sydney, NSW, 2001
?

**Adelaide Atari Computer Club**
Neil Patterson, President
P.O. Box 333
Kent Town, SA, 5071
8-Bit, ST

**Burnie Atari Computer Club**
Alex Bienefelt
P.O. Box 99
Ridgley, TAS, 7321
?

**Melbourne Atari Computer Enthusiasts**
MACE
Roger Davies, President
P.O. Box 340
Rosanna, VIC, 3084
Mostly ST

**Queensland Atari Computer Enthusiasts**
Peter Wighton, President
P.O. Box 10026
Brisbane, OLD, 4000
?

**W.A. Atari & IBM Users Club**
Colin Hunt, President
19 Wandarrie Avenue
Yokine, W.A., 6060
?

## No Time Like The Present — Subscriptions/Renewals

Name:_____

Address:_____

City:_____ State/Country:_____

ZIP/Postal Code:_____ Phone(Optional):_____

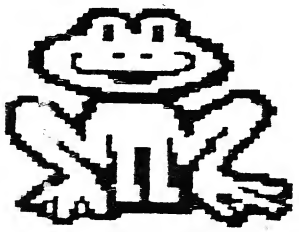
### Rates

| | 6 Issues | 12 Issues | 18 Issues |
|---|---|---|---|
| USA 2nd Class Mail | [ ] $25 | [ ] $46 | [ ] $63 |
| Canada Surface Mail | [ ] $30 | [ ] $56 | [ ] $78 |
| Other Foreign Surface Mail | [ ] $32 | [ ] $60 | [ ] $84 |
| European Airmail | [ ] $38 | [ ] $72 | [ ] $102 |
| Asia/Pacific Airmail | [ ] $40 | [ ] $76 | [ ] $108 |
| AC Software Disk (Anywhere) | [ ] $9 | [ ] $18 | [ ] $27 |

Published no more than six times a year.
Software disks will be distributed every second issue.
Magazines mailed in plain envelopes to arrive at your doorstep fresh and clean!

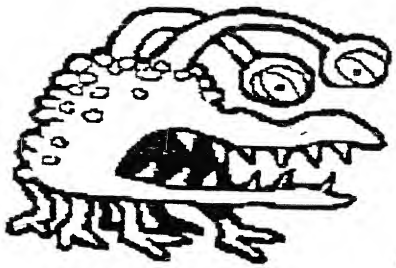
**ATARI CLASSICS, 1161 BAY STREET, ALAMEDA, CA 94501**

VIDEO 61
22735 CONGO ST NE
STACY MN 55079
612-462-2500



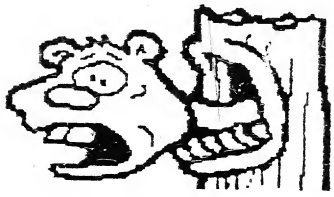
WE SPECIALIZE IN THE ATARI 2600,5200,7800,8-BIT COMPUTERS,LYNX AND THE JAGUAR
CALL US FOR CARTRIDGES, DISKS,POWER SUPPLIES,SWITCH BOXES,AND MORE!
WE EVEN REBUILD YOUR 5200 JOYSTICKS!!

SOFTWARE NEW IN BOX
200 TITLES FOR 2600
35 TITLES FOR 5200
50 TITLES FOR 7800
300 TITLES FOR THE 8-BIT COMPUTERS
72 TITLES FOR THE LYNX
FOR THE JAGUAR CALL



SOME OF THE STUFF THAT WE CARRY FOR THE ATARI IS OUT OF THIS WORLD!!

OTHER STUFF IS JUST PLAIN WACKY!!



THIS WAS DONE WITH NEWSROOM FOR THE ATARI 8-BIT COMPUTER.


# MORE THAN GAMES

Classic 8 Bit

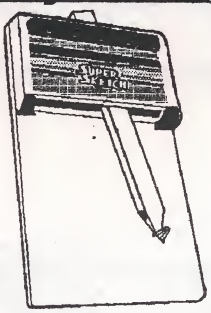
## Productivity

| | | | |
|---|---|---|---|
| Home Filing Manager | 2.00 | Family Finances | 1.50 |
| Timewise | 2.00 | Elementary Atari Book | 2.50 |
| Atari BookKeeper | 2.50 | | |

LOOSE 1.00 ea.: Atari BASIC rev.'A' cart

Coming Soon: Rix MicroDesk

## Super Sketch Graphics Tablet



**Slip a 1/2 letter size drawing onto the tablet and use the stylus to begin tracing, or draw freehand. Four colors, patterns, fills, custom/grab pattern, save/load, cut/paste, text, brushes, magnify, and tools. With ROM paint cartridge.**

***Now with convert & print software. ver:1.0!***

**$6.00 ea.,(or buy a case of 12 for $60.00, postage paid.)**

## Games

| | | | |
|---|---|---|---|
| Match Wits | 1.00 | Silent Service | 1.25 |
| F15 Strike Eagle | 1.25 | Drop Zone | 4.00 |
| Bandits | 1.00 | Ballblazer | 2.00 |
| Summer Games | 0.75 | | |

Infocom: Moonmist or Hollywood Hijinx 3.00

LOOSE 1.00 ea.: Space Invaders, Missle Command, Math Encounter

## Hardware

| | | | |
|---|---|---|---|
| Atari Trackball | 6.00 | Composite monitor cable | 3.00 |
| SIO Cable 3 ft. | 1.00 | Dix cass dual cleaner | 1.00 |
| Joystick port extender | 10.00 | | |
| 800 computer 48K,BASIC,,PWR | 25.00 | 800/1050 power supply | 5.00 |
| ATR-8000 Interface | 65.00 | 850 Interface | 30.00 |

Coming Soon: Rix SpeedIO

## and Other

| | | | |
|---|---|---|---|
| 5" New DSDD disks | 10/1.00 | 5" NEW 1.2meg disks | 10/1.50 |
| 3.5" USED 720 disks | 10/1.50 | 5" Disk Notcher | 1.25 |
| LX80 Printer Ribbon | 0.50 | Citizen 120D Ribbon | 2.00 |
| Paddles(used) | 2.50 | Null Modem Adapter | 6.30 |
| Joystick Port Extend. | 10.00 | 850-DB25 Ptr adapter | 10.00 |

1020 White Paper Roll: Sgl 1.00, DBL 1.50(coming soon:YELLOW paper)
Centronics Printer Cable 6' 4.25
BOOKS 1.00 each: 130XE; DOS 2.5, XM301, Computers for People, Hacker Prog/Tips, 400/800 ref..
And we keep a supply of generic parts on hand.

MAGAZINES:We have hundreds of Antics and Analogs to choose from, and many other titles too. Mags alone, with disks, or with copies(as available). $1.00 / $1.50 / $3.50. Request list.

**MANY MORE 1 ONLYS.**

Our flyer has many things for the ST system and the 2600 game system, among others.
We can make custom cables and cases.
We can do repairs and upgrades.

POSTAGE is $6.00 for continental U.S. deliveries.
Quantities Limited. Prices and Advert subject to change.
Items sold AS IS, replaced if defective, if possible.
(TX addresses add 8 % sales tax)
New and Previously owned items arriving frequently.
For Flyer requests or Orders, Write to:


**More Than Games**
**8207 Briarwood Lane**
**Austin, TX 78757-7642**
**Email Info:74766.1561@compuserve.com**


Nearly everything in our flyer is 30-80% off our regular low prices. We are closing out the resale side.
Postmark orders by ~~Feb.28, 1997.~~
(* March 15th, 97 for AC Readers!*)
See you at the Feb. 22 Houston Show.



**** CLEARANCE SALE ****